20-COMM-E EtherNet/IP Adapter

Allen-Bradley

Series A FRN 2.xxx
Series B FRN 4.xxx

User Manual

Allen-Bradley • Rockwell Software

Rockwell Automation

Chapter 6

Using Explicit Messaging

This chapter provides information and examples that explain how to use Explicit Messaging to configure and monitor the adapter and connected PowerFlex 7-Class drive or PowerFlex 750-Series drive.

Important: When used in a PowerFlex 750-Series drive, the 20-COMM-E adapter must have firmware version 4.001 (or later) to support explicit messaging to drive parameters (Port 0). Furthermore, the adapter requires firmware version 4.002 (or later) for explicit messaging to parameters of peripherals in drive Ports 1…14.

⚠ **ATTENTION:** Risk of injury or equipment damage exists. The examples in this publication are intended solely for purposes of example. There are many variables and requirements with any application. Rockwell Automation, Inc. does not assume responsibility or liability (to include intellectual property liability) for actual use of the examples shown in this publication.

⚠ **ATTENTION:** Risk of equipment damage exists. If Explicit Messages are programmed to write parameter data to Non-Volatile Storage (NVS) frequently, the NVS will quickly exceed its life cycle and cause the drive to malfunction. Do not create a program that frequently uses Explicit Messages to write parameter data to NVS. Datalinks do not write to NVS and should be used for frequently changed parameters.

Refer to Chapter 5 for information about the I/O Image, using Logic Command/Status, Reference/Feedback, and Datalinks.

About Explicit Messaging

Explicit Messaging is used to transfer data that does not require continuous updates. With Explicit Messaging, you can configure and monitor a slave device's parameters on the network.

Important: When an explicit message is performed, by default no connection is made since it is an "unconnected" message. When timing of the message transaction is important, you can create a dedicated message connection between the controller and drive by checking the "Connected" box on the Communications tab message configuration screen during message setup. These message connections are in addition to the I/O connection. However, the trade off for more message connections is decreased network performance. If your application cannot tolerate this, do not check the "Connected" box, which is recommended.

Important: PowerFlex 7-Class and PowerFlex 750-Series drives have explicit messaging limitations. Table 6.A shows the EtherNet/IP Object Class code compatibilities for these drives.

Table 6.A Explicit Messaging Class Code Compatibility with Drives

EtherNet/IP Object Class Code	PowerFlex 7-Class Drives	PowerFlex 750-Series Drives	Explicit Messaging Function
Parameter Object 0x0F	Yes	No	Single parameter reads/writes
DPI Parameter Object 0x93	Yes	Yes[1] with limitations	Single and scattered parameter reads/writes
Host DPI Parameter Object 0x9F	No	Yes[2] with limitations	Single and scattered parameter reads/writes

[1] Enables access to drive parameters (Port 0), DPI device parameters (Ports 1…6 only), and Host parameters (Ports 7…14 only). For example, DPI Parameter Object Class code 0x93 can access a 20-COMM-E adapter in Port 6. However, Class code 0x93 cannot access, for example, the Host parameters in a 24V I/O option module in Port 5. See DPI Parameter Object on page C-16 for instance (parameter) numbering.

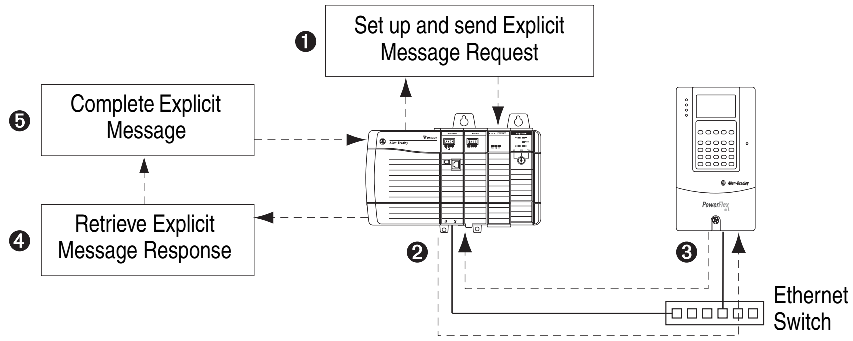
[2] Enables access to drive parameters (Port 0) and Host parameters for all ports (1…14). Host DPI Parameter Object Class code 0x9F cannot access DPI (device) parameters. For example, if a 20-750-DNET option module is in Port 4, its Host parameters can be accessed, but not its DPI (device) parameters. See Host DPI Parameter Object on page C-30 for instance (parameter) numbering.

Performing Explicit Messages

There are five basic events in the Explicit Messaging process. The details of each step will vary depending on the type of controller being used. Refer to the documentation for your controller.

Important: There must be a request message and a response message for all Explicit Messages, whether you are reading or writing data.

Figure 6.1 Explicit Message Process


❶
Set up and send Explicit Message Request
❺
Complete Explicit Message
❹
Retrieve Explicit Message Response
❷
❸
Ethernet Switch

Event	Description
❶	You format the required data and set up the ladder logic program to send an Explicit Message request to the scanner or bridge module (download).
❷	The scanner or bridge module transmits the Explicit Message Request to the slave device over the network.
❸	The slave device transmits the Explicit Message Response back to the scanner. The data is stored in the scanner buffer.
❹	The controller retrieves the Explicit Message Response from the scanner's buffer (upload).
❺	The Explicit Message is complete.

For information on the maximum number of Explicit Messages that can be executed at a time, refer to the user manual for the scanner or bridge and/or controller that is being used.

ControlLogix Examples

TIP: To display the Message Configuration screen in RSLogix 5000, add a message instruction (MSG), create a new tag for the message (Properties: Base tag type, MESSAGE data type, controller scope), and click the ... button in the message instruction.

For supported classes, instances, and attributes, refer to Appendix C, EtherNet/IP Objects.

Explicit Messaging Using RSLogix 5000 Version 15 (or later)

ControlLogix Example Ladder Logic Program to Read a Single Parameter

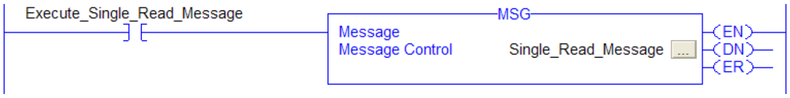
A Parameter Read message is used to read a single parameter. This read message example reads the value of parameter 003 - [Output Current] in a PowerFlex 7-Class drive.

Important: Parameter Object Class code 0x0F is not supported in PowerFlex 750-Series drives. To do a single parameter read, follow the RSLogix 5000 (all versions) single read example on page 6-19.

Table 6.B Example Controller Tags to Read a Single Parameter

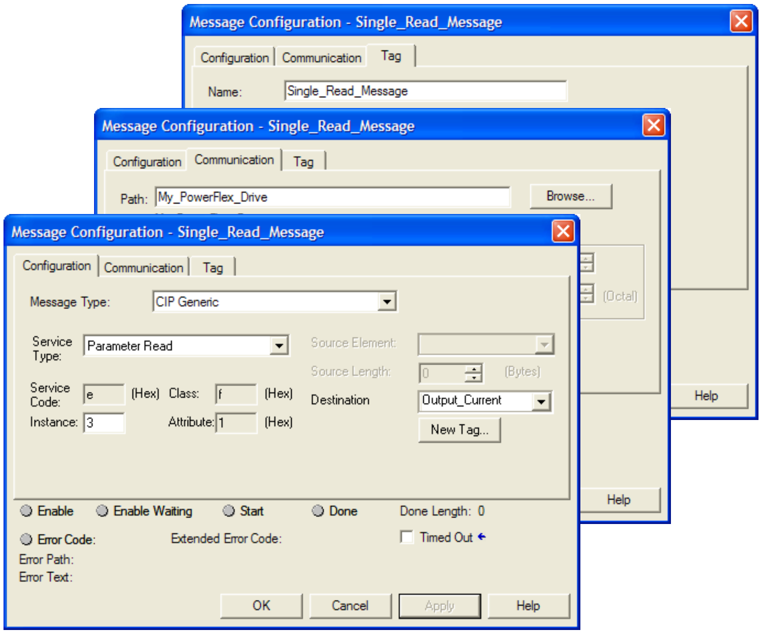
Operand	Controller Tags for Single Read Message	Data Type
XIC	Execute_Single_Read_Message	BOOL
MSG	Single_Read_Message	MESSAGE

Figure 6.2 Example Ladder Logic to Read a Single Parameter



ControlLogix – Formatting a Message to Read a Single Parameter (version 15 or later)

Figure 6.3 Parameter Read Single Message Configuration Screens



Message Configuration - Single_Read_Message

Tag tab: Name: Single_Read_Message

Communication tab: Path: My_PowerFlex_Drive

Configuration tab:

Message Type: CIP Generic

Service Type: Parameter Read

Service Code: e (Hex) Class: f (Hex)

Instance: 3 Attribute: 1 (Hex)

Source Element:

Source Length: 0 (Bytes)

Destination: Output_Current

○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0

○ Error Code: Extended Error Code: □ Timed Out

Error Path:

Error Text:

The following table identifies the data that is required in each box to configure a message to read a single parameter.

Configuration Tab	Example Value	Description
Message Type Service Type [1] Service Code [1] Class Instance [2] Attribute Destination	CIP Generic Parameter Read e (Hex.) f (Hex.) 3 (Dec.) 1 (Hex.) Output_Current [4]	Used to access the Parameter Object in the adapter. This service is used to read a parameter value. Code for the requested service. Class ID for the DPI Parameter Object. Instance number is the same as parameter number. Attribute number for the Parameter Value attribute. The tag where the data that is read is stored.
Communication Tab	**Example Value**	**Description**
Path [3]	My_PowerFlex_Drive	The path is the route that the message will follow.
Tag Tab	**Example Value**	**Description**
Name	Single_Read_Message	The name for the message.

[1] The default setting for Service Type is "Custom," enabling entry of a Service Code not available from the Service Type pull-down menu. When choosing a Service Type other than "Custom" from the pull-down menu, an appropriate Hex. value is automatically assigned to the Service Code box which is dimmed (unavailable).

[2] Only drive parameters (Port 0) can be read using Parameter Object Class code 0x0F. To read a parameter of a peripheral in another port, use DPI Parameter Object Class code 0x93 (see page 6-19).

[3] Click **Browse** to find the path, or type in the name of the device listed in the I/O Configuration folder.

[4] In this example, Output Current is a 32-bit parameter requiring the Data Type field to be set to "DINT" when creating the controller tag. If the parameter being read is a 16-bit parameter, the tag Data Type field must be set to "INT." When using a PowerFlex 700S drive, Output Current is a floating point parameter requiring the Data Type field to be set to "REAL" when creating the controller tag. See the drive documentation to determine the size of the parameter and its data type (16-bit or 32-bit integer or REAL).

ControlLogix Example Ladder Logic Program to Write a Single Parameter (version 15 or later)

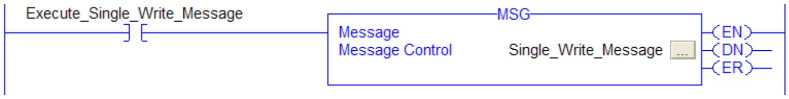
A Parameter Write message is used to write to a single parameter. This write message example writes a value to parameter 140 - [Accel Time 1] in a PowerFlex 7-Class drive.

Important: Parameter Object Class code 0x0F is not supported in PowerFlex 750-Series drives. To do a single parameter write, follow the RSLogix 5000 (all versions) single write example on page 6-21.

Table 6.C Example Controller Tags to Write a Single Parameter

Operand	Controller Tags for Single Write Message	Data Type
XIC	Execute_Single_Write_Message	BOOL
MSG	Single_Write_Message	MESSAGE

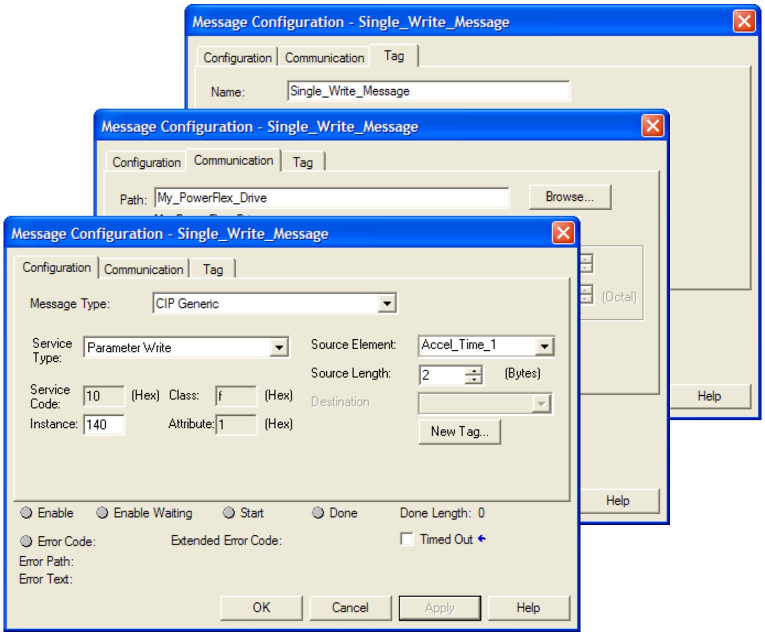
Figure 6.4 Example Ladder Logic to Write a Single Parameter



Important: If the explicit message single write must be written continuously, use DPI Parameter Object Class code 0x93 or Host DPI Parameter Object Class code 0x9F and attribute A (10 decimal; see page 6-22). This writes to RAM—not NVS (EEPROM) memory. This example single write message using Class code F writes to NVS. Over time, continuous writes will exceed the EEPROM life cycle and cause the drive to malfunction.

ControlLogix – Formatting a Message to Write a Single Parameter (version 15 or later)

Figure 6.5 Parameter Write Single Message Configuration Screens



Message Configuration - Single_Write_Message

Name: Single_Write_Message

Path: My_PowerFlex_Drive

Message Type: CIP Generic

Service Type: Parameter Write

Service Code: 10 (Hex) Class: f (Hex)

Instance: 140 Attribute: 1 (Hex)

Source Element: Accel_Time_1

Source Length: 2 (Bytes)

Destination

○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0

○ Error Code: Extended Error Code: □ Timed Out

Error Path:

Error Text:

The following table identifies the data that is required in each box to configure a message to write a single parameter.

Configuration Tab	Example Value	Description
Message Type	CIP Generic	Used to access the Parameter Object in the adapter.
Service Type [1]	Parameter Write	This service is used to write a parameter value.
Service Code [1]	10 (Hex.)	Code for the requested service.
Class	f (Hex.)	Class ID for the DPI Parameter Object.
Instance [2]	140 (Dec.)	Instance number is the same as parameter number.
Attribute	1 (Hex.)	Attribute number for the Parameter Value attribute.
Source Element	Accel_Time_1 [4]	Name of the tag for any service data to be sent from the scanner or bridge to the adapter/drive.
Source Length	2 [4]	Number of bytes of service data to be sent in the message.
Communication Tab	**Example Value**	**Description**
Path [3]	My_PowerFlex_Drive	The path is the route that the message will follow.
Tag Tab	**Example Value**	**Description**
Name	Single_Write_Message	The name for the message.

[1] The default setting for Service Type is "Custom," enabling entry of a Service Code not available from the Service Type pull-down menu. When choosing a Service Type other than "Custom" from the pull-down menu, an appropriate Hex. value is automatically assigned to the Service Code box which is dimmed (unavailable).

[2] Only drive parameters (Port 0) can be written to using Parameter Object Class code 0x0F. To write to a parameter of a peripheral in another port, use DPI Parameter Object Class code 0x93 (see page 6-21).

[3] Click **Browse** to find the path, or type in the name of the device listed in the I/O Configuration folder.

[4] In this example, Accel Time 1 is a 16-bit parameter requiring the tag Data Type field to be set to "INT" when creating the controller tag. If the parameter being written to is a 32-bit parameter, the tag Data Type field must be set to "DINT." Also, the Source Length field on the Message Configuration screen must correspond to the selected Data Type in bytes (for example, 4 bytes for a DINT or a REAL). When using a PowerFlex 700S drive, Accel Time 1 is a floating point number requiring the Data Type field to be set to "REAL" when creating the controller tag. See the drive documentation to determine the size of the parameter and its data type (16-bit or 32-bit integer or REAL).

ControlLogix Example Ladder Logic Program to Read Multiple Parameters (all versions)

A Scattered Read message is used to read the values of multiple parameters. Up to 22 parameters can be read. This read message example reads the values of these five parameters:

PowerFlex 7-Class Drive	PowerFlex 750-Series Drive
• Parameter 001 - [Output Freq]	• Parameter 001 - [Output Freq]
• Parameter 003 - [Output Current]	• Parameter 007 - [Output Current]
• Parameter 006 - [Output Voltage]	• Parameter 137 - [Open Loop Fdbk]
• Parameter 012 - [DC Bus Voltage]	• Parameter 21581 - [Port 5: Analog Out 0 Data]
• Parameter 017 - [Analog In1 Value]	• Parameter 260 - [Analog In0 Value]

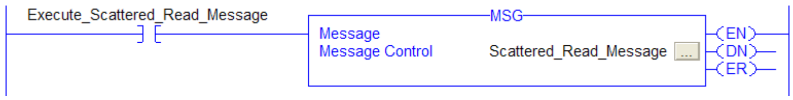
Refer to DPI Parameter Object on page C-16 (Class code 0x93) or Host DPI Parameter Object on page C-30 (Class code 0x9F) for parameter numbering.

Important: See Table 6.A on page 6-2 for limitations of PowerFlex 7-Class and PowerFlex 750-Series drives when using Class code 0x93 or Class code 0x9F for explicit messaging.

Table 6.D Example Controller Tags to Read Multiple Parameters

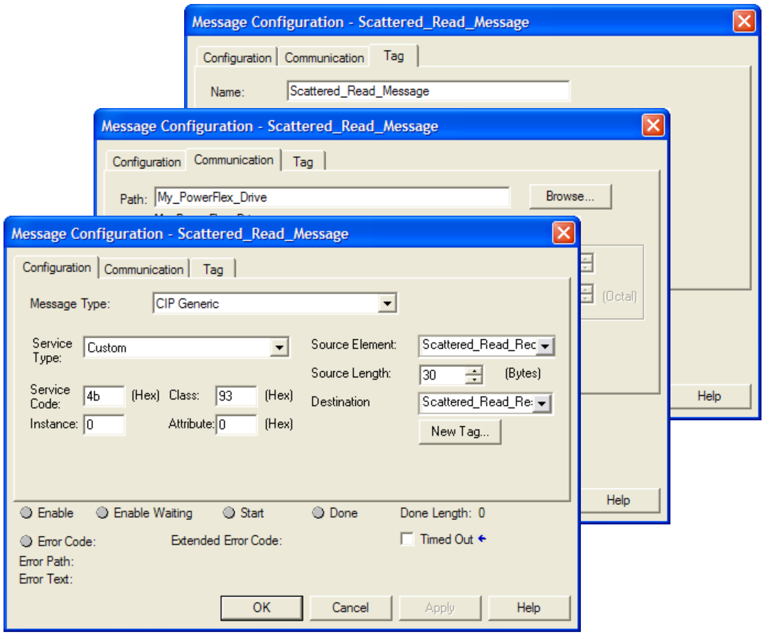
Operand	Controller Tags for Read Multiple Message	Data Type
XIC	Execute_Scattered_Read_Message	BOOL
MSG	Scattered_Read_Message	MESSAGE

Figure 6.6 Example Ladder Logic to Read Multiple Parameters



ControlLogix – Formatting a Message to Read Multiple Parameters (all versions)

Figure 6.7 Scattered Read Message Configuration Screens



Message Configuration - Scattered_Read_Message

Name: Scattered_Read_Message

Message Configuration - Scattered_Read_Message

Path: My_PowerFlex_Drive

Message Configuration - Scattered_Read_Message

Message Type: CIP Generic

Service Type: Custom

Service Code: 4b (Hex) Class: 93 (Hex)

Instance: 0 Attribute: 0 (Hex)

Source Element: Scattered_Read_Rec

Source Length: 30 (Bytes)

Destination Scattered_Read_Re:

○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0

○ Error Code: Extended Error Code: □ Timed Out

Error Path:

Error Text:

The following table identifies the data that is required in each box to configure a message to read multiple parameters.

Configuration Tab	**Example Value**	**Description**
Message Type	CIP Generic	Used to access Parameter Object in the adapter.
Service Type [1]	Custom	Required for scattered messages.
Service Code [1]	4b (Hex.)	Code for the requested service.
Class	93 or 9F (Hex.) [3]	Class ID for the DPI Parameter Object.
Instance	0 (Dec.)	Required for scattered messages.
Attribute	0 (Hex.)	Required for scattered messages.
Source Element	Scattered_Read_Request [4]	Name of the tag for any service data to be sent from scanner or bridge to the adapter/drive.
Source Length	30 [4]	Number of bytes of service data to be sent in the message.
Destination	Scattered_Read_Response	The tag where the data that is read is stored.
Communication Tab	**Example Value**	**Description**
Path [2]	My_PowerFlex_Drive	The path is the route that the message will follow.
Tag Tab	**Example Value**	**Description**
Name	Scattered_Read_Message	The name for the message.

[1] The default setting for Service Type is "Custom," enabling entry of a Service Code not available from the Service Type pull-down menu. When choosing a Service Type other than "Custom" from the pull-down menu, an appropriate Hex. value is automatically assigned to the Service Code box which is dimmed (unavailable).

[2] Click **Browse** to find the path, or type in the name of the device listed in the I/O Configuration folder.

[3] See Table 6.A on page 6-2 for limitations of PowerFlex 7-Class and PowerFlex 750-Series drives when using DPI Parameter Object Class code 0x93 or Host DPI Parameter Object Class code 0x9F for explicit messaging.

[4] In this example, we are reading five parameters. Each parameter being read requires an array of three INT registers. Therefore, a controller tag was created with its Data Type field set to "INT[15]." Also, the Source Length field on the Message Configuration screen must correspond to the selected Data Type in bytes (for this example, 30 bytes for an INT[15] array). Scattered read messages always assume that every parameter being read is a 32-bit parameter, regardless of its actual size. Maximum length is 132 bytes or 66 words which equates to 22 parameters. For parameter numbering, see DPI Parameter Object on page C-16 (Class code 0x93) or Host DPI Parameter Object on page C-30 (Class code 0x9F).

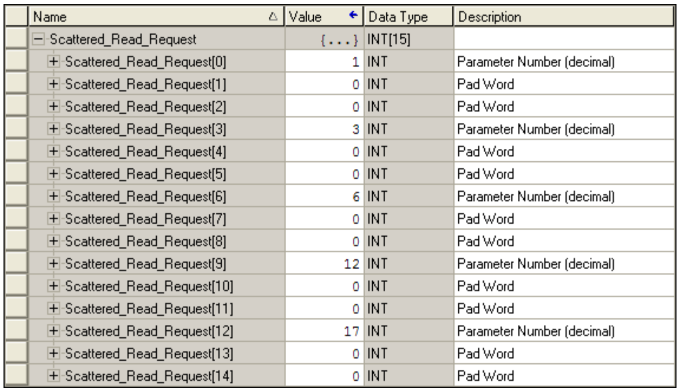
ControlLogix Example Scattered Read Request Data

In this message example, we use the data structure in Figure 6.8 or Figure 6.9 in the source tag named Scattered_Read_Request to read these five parameters:

PowerFlex 7-Class Drive	PowerFlex 750-Series Drive
• Parameter 001 - [Output Freq]	• Parameter 001 - [Output Freq]
• Parameter 003 - [Output Current]	• Parameter 007 - [Output Current]
• Parameter 006 - [Output Voltage]	• Parameter 137 - [Open Loop Fdbk]
• Parameter 012 - [DC Bus Voltage]	• Parameter 21581 - [Port 5: Analog Out 0 Data]
• Parameter 017 - [Analog In1 Value]	• Parameter 260 - [Analog In0 Value]

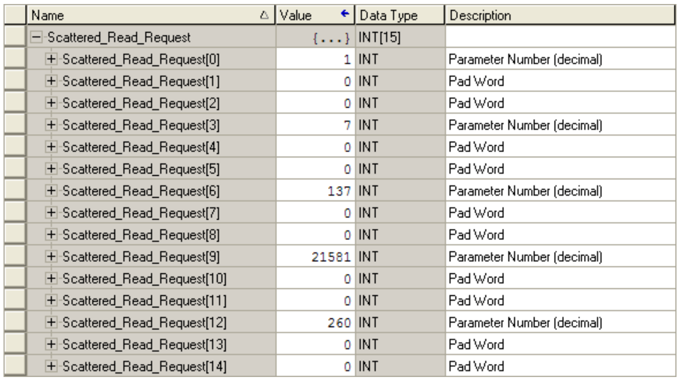
Refer to DPI Parameter Object on page C-16 (Class code 0x93) or Host DPI Parameter Object on page C-30 (Class code 0x9F) for parameter numbering.

Figure 6.8 Example Scattered Read Request Data for PowerFlex 7-Class Drive



Name	Value	Data Type	Description
Scattered_Read_Request	{...}	INT[15]	
Scattered_Read_Request[0]	1	INT	Parameter Number (decimal)
Scattered_Read_Request[1]	0	INT	Pad Word
Scattered_Read_Request[2]	0	INT	Pad Word
Scattered_Read_Request[3]	3	INT	Parameter Number (decimal)
Scattered_Read_Request[4]	0	INT	Pad Word
Scattered_Read_Request[5]	0	INT	Pad Word
Scattered_Read_Request[6]	6	INT	Parameter Number (decimal)
Scattered_Read_Request[7]	0	INT	Pad Word
Scattered_Read_Request[8]	0	INT	Pad Word
Scattered_Read_Request[9]	12	INT	Parameter Number (decimal)
Scattered_Read_Request[10]	0	INT	Pad Word
Scattered_Read_Request[11]	0	INT	Pad Word
Scattered_Read_Request[12]	17	INT	Parameter Number (decimal)
Scattered_Read_Request[13]	0	INT	Pad Word
Scattered_Read_Request[14]	0	INT	Pad Word

Figure 6.9 Example Scattered Read Request Data for PowerFlex 750-Series Drive

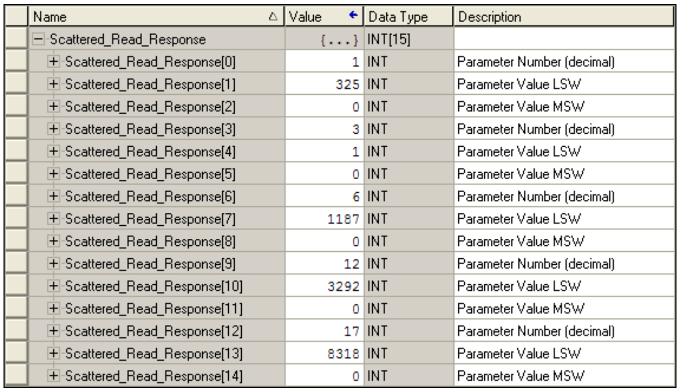


Name	Value	Data Type	Description
Scattered_Read_Request	{...}	INT[15]	
Scattered_Read_Request[0]	1	INT	Parameter Number (decimal)
Scattered_Read_Request[1]	0	INT	Pad Word
Scattered_Read_Request[2]	0	INT	Pad Word
Scattered_Read_Request[3]	7	INT	Parameter Number (decimal)
Scattered_Read_Request[4]	0	INT	Pad Word
Scattered_Read_Request[5]	0	INT	Pad Word
Scattered_Read_Request[6]	137	INT	Parameter Number (decimal)
Scattered_Read_Request[7]	0	INT	Pad Word
Scattered_Read_Request[8]	0	INT	Pad Word
Scattered_Read_Request[9]	21581	INT	Parameter Number (decimal)
Scattered_Read_Request[10]	0	INT	Pad Word
Scattered_Read_Request[11]	0	INT	Pad Word
Scattered_Read_Request[12]	260	INT	Parameter Number (decimal)
Scattered_Read_Request[13]	0	INT	Pad Word
Scattered_Read_Request[14]	0	INT	Pad Word

ControlLogix Example Scattered Read Response Data

The Scattered Read Request message reads the multiple parameters and returns their values to the destination tag (Scattered_Read_Response). Figure 6.10 or Figure 6.11 shows the parameter values.

Figure 6.10 Example Scattered Read Response Data for PowerFlex 7-Class Drive

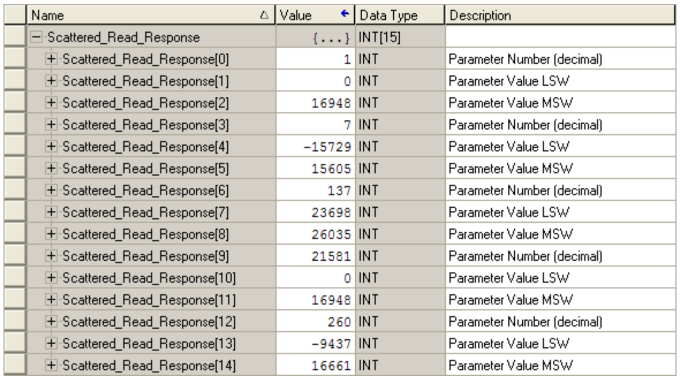


Name	Value	Data Type	Description
⊟ Scattered_Read_Response	{...}	INT[15]	
⊞ Scattered_Read_Response[0]	1	INT	Parameter Number (decimal)
⊞ Scattered_Read_Response[1]	325	INT	Parameter Value LSW
⊞ Scattered_Read_Response[2]	0	INT	Parameter Value MSW
⊞ Scattered_Read_Response[3]	3	INT	Parameter Number (decimal)
⊞ Scattered_Read_Response[4]	1	INT	Parameter Value LSW
⊞ Scattered_Read_Response[5]	0	INT	Parameter Value MSW
⊞ Scattered_Read_Response[6]	6	INT	Parameter Number (decimal)
⊞ Scattered_Read_Response[7]	1187	INT	Parameter Value LSW
⊞ Scattered_Read_Response[8]	0	INT	Parameter Value MSW
⊞ Scattered_Read_Response[9]	12	INT	Parameter Number (decimal)
⊞ Scattered_Read_Response[10]	3292	INT	Parameter Value LSW
⊞ Scattered_Read_Response[11]	0	INT	Parameter Value MSW
⊞ Scattered_Read_Response[12]	17	INT	Parameter Number (decimal)
⊞ Scattered_Read_Response[13]	8318	INT	Parameter Value LSW
⊞ Scattered_Read_Response[14]	0	INT	Parameter Value MSW

In this message example, the parameters have the following values:

PowerFlex 7-Class Drive Parameter	Read Value
1 - [Output Freq]	32.5 Hz
3 - [Output Current]	0.01 Amp
6 - [Output Voltage]	118.7 VAC
12 - [DC Bus Voltage]	329.2 VDC
17 - [Analog In2 Value]	8.318 mA

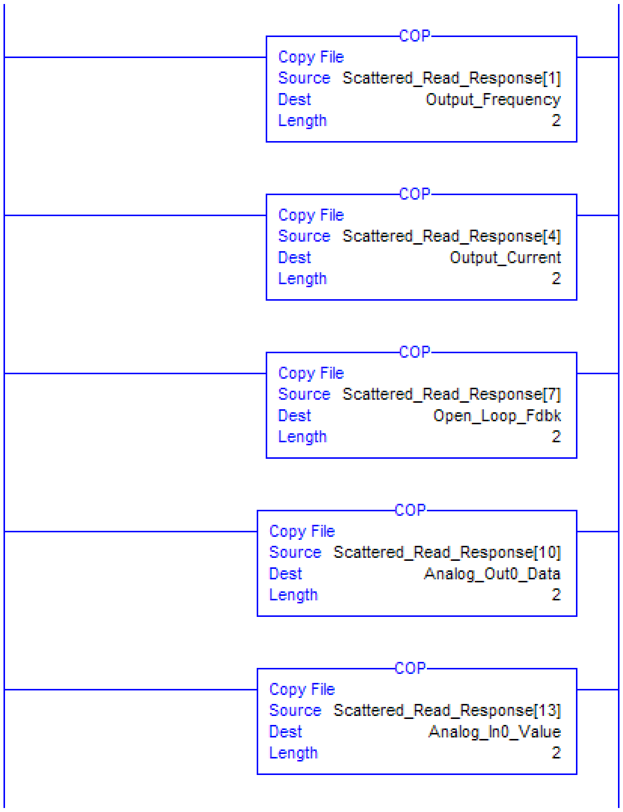
Figure 6.11 Example Scattered Read Response Data for PowerFlex 750-Series Drive



Name	Value	Data Type	Description
⊟ Scattered_Read_Response	{...}	INT[15]	
⊞ Scattered_Read_Response[0]	1	INT	Parameter Number (decimal)
⊞ Scattered_Read_Response[1]	0	INT	Parameter Value LSW
⊞ Scattered_Read_Response[2]	16948	INT	Parameter Value MSW
⊞ Scattered_Read_Response[3]	7	INT	Parameter Number (decimal)
⊞ Scattered_Read_Response[4]	-15729	INT	Parameter Value LSW
⊞ Scattered_Read_Response[5]	15605	INT	Parameter Value MSW
⊞ Scattered_Read_Response[6]	137	INT	Parameter Number (decimal)
⊞ Scattered_Read_Response[7]	23698	INT	Parameter Value LSW
⊞ Scattered_Read_Response[8]	26035	INT	Parameter Value MSW
⊞ Scattered_Read_Response[9]	21581	INT	Parameter Number (decimal)
⊞ Scattered_Read_Response[10]	0	INT	Parameter Value LSW
⊞ Scattered_Read_Response[11]	16948	INT	Parameter Value MSW
⊞ Scattered_Read_Response[12]	260	INT	Parameter Number (decimal)
⊞ Scattered_Read_Response[13]	-9437	INT	Parameter Value LSW
⊞ Scattered_Read_Response[14]	16661	INT	Parameter Value MSW

The PowerFlex 750-Series drive uses 32-bit integer and REAL parameters. A COP command must be used to copy the Scattered_Read_Response integer array to a 32-bit integer or REAL tag. Figure 6.12 shows the ladder logic used for this example. If the parameter data type is a REAL, then the destination tag is a REAL. If the parameter data type is a 32-bit integer, then the destination tag is a DINT. See the drive documentation to determine the parameter data type (32-bit integer or REAL).

Figure 6.12 Example Ladder Logic to Copy Response Data for PowerFlex 750-Series Drive



```
COP
Copy File
Source  Scattered_Read_Response[1]
Dest                Output_Frequency
Length                             2

COP
Copy File
Source  Scattered_Read_Response[4]
Dest                  Output_Current
Length                             2

COP
Copy File
Source  Scattered_Read_Response[7]
Dest                  Open_Loop_Fdbk
Length                             2

COP
Copy File
Source  Scattered_Read_Response[10]
Dest                  Analog_Out0_Data
Length                             2

COP
Copy File
Source  Scattered_Read_Response[13]
Dest                  Analog_In0_Value
Length                             2
```

In this message example, the parameters have the following values:

PowerFlex 750-Series Drive Parameter	Read Value	Data Type
1 - [Output Freq]	45.0 Hz	REAL
7 - [Output Current]	0.03 Amp	REAL
137 - [Open Loop Fdbk]	1706253458	DINT
21581 - [Port 5: Analog Out 0 Data]	45.0 Hz	REAL
260 - [Analog In0 Value]	9.366 Volts	REAL

ControlLogix Example Ladder Logic Program to Write Multiple Parameters (all versions)

A Scattered Write message is used to write to multiple parameters. This write message example writes the following values to these five parameters:

PowerFlex 7-Class Drive Parameter	Write Value
141 - [Accel Time 2]	11.1 Sec.
143 - [Decel Time 2]	22.2 Sec.
105 - [Preset Speed 5]	33.3 Hz.
106 - [Preset Speed 6]	44.4 Hz.
107 - [Preset Speed 7]	55.5 Hz.

PowerFlex 750-Series Drive Parameter	Write Value
536 - [Accel Time 2]	11.1 Sec.
538 - [Decel Time 2]	22.2 Sec.
725 - [Zero Position]	33
21555 - [Port 5: Analog In0 Hi]	5.5
780 - [PTP Setpoint]	-75,555

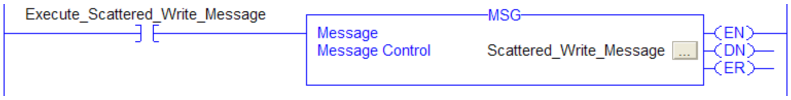
Refer to DPI Parameter Object on page C-16 (Class code 0x93) or Host DPI Parameter Object on page C-30 (Class code 0x9F) for parameter numbering.

Important: See Table 6.A on page 6-2 for limitations of PowerFlex 7-Class and PowerFlex 750-Series drives when using Class code 0x93 or Class code 0x9F for explicit messaging.

Table 6.E Example Controller Tags to Write Multiple Parameters

Operand	Controller Tags for Write Multiple Message	Data Type
XIC	Execute_Scattered_Write_Message	BOOL
MSG	Scattered_Write_Message	MESSAGE

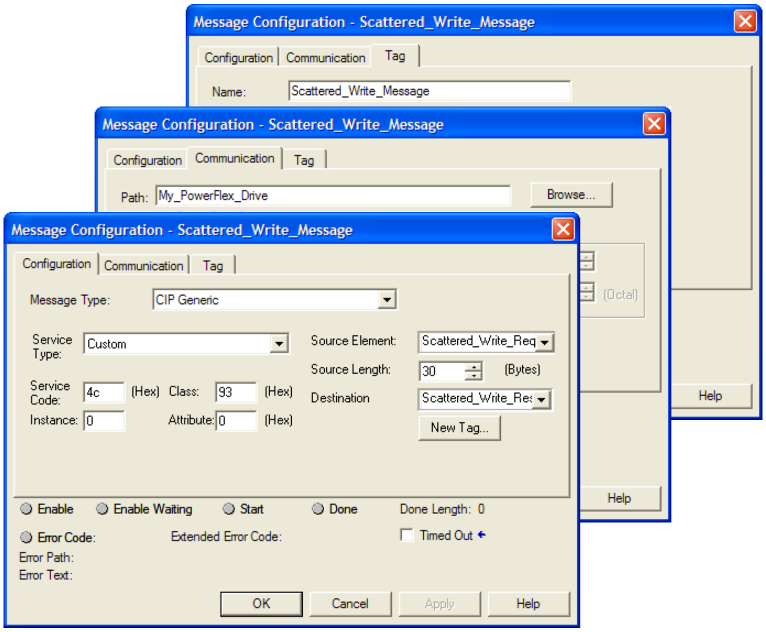
Figure 6.13 Example Ladder Logic to Write Multiple Parameters



Important: If the explicit message scattered write must be written continuously, then use a separate explicit message single write for each parameter using DPI Parameter Object Class code 0x93 or Host DPI Parameter Object Class code 0x9F and attribute A (10 decimal; see page 6-22). Attribute A writes to RAM—not NVS (EEPROM) memory. This example scattered write message using attribute 0 writes to NVS. Over time, continuous writes will exceed the EEPROM life cycle and cause the drive to malfunction.

ControlLogix – Formatting a Message to Write Multiple Parameters (all versions)

Figure 6.14 Scattered Write Multiple Message Configuration Screens



Name: Scattered_Write_Message

Path: My_PowerFlex_Drive

Message Type: CIP Generic

Service Type: Custom

Service Code: 4c (Hex) Class: 93 (Hex)

Instance: 0 Attribute: 0 (Hex)

Source Element: Scattered_Write_Req

Source Length: 30 (Bytes)

Destination: Scattered_Write_Res

○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0

○ Error Code: Extended Error Code: □ Timed Out

Error Path:

Error Text:

The following table identifies the data that is required in each box to configure a message to write multiple parameters.

Configuration Tab	Example Value	Description
Message Type Service Type [1] Service Code [1] Class Instance Attribute Source Element Source Length Destination	CIP Generic Custom 4c (Hex.) 93 or 9F (Hex.) [3] 0 (Dec.) 0 (Hex.) Scattered_Write_Request [4] 30 [4] Scattered_Write_Response	Used to access Parameter Object in the adapter. Required for scattered messages. Code for the requested service. Class ID for the DPI Parameter Object. Required for scattered messages. Required for scattered messages. Name of the tag for any service data to be sent from scanner or bridge to the adapter/drive. Number of bytes of service data to be sent in the message. The tag where the data that is read is stored.
Communication Tab	**Example Value**	**Description**
Path [2]	My_PowerFlex_Drive	The path is the route that the message will follow.
Tag Tab	**Example Value**	**Description**
Name	Scattered_Write_Message	The name for the message.

[1] The default setting for Service Type is “Custom,” enabling entry of a Service Code not available from the Service Type pull-down menu. When choosing a Service Type other than “Custom” from the pull-down menu, an appropriate Hex. value is automatically assigned to the Service Code box which is dimmed (unavailable).

[2] Click **Browse** to find the path, or type in the name of the device listed in the I/O Configuration folder.

[3] See Table 6.A on page 6-2 for limitations of PowerFlex 7-Class and PowerFlex 750-Series drives when using DPI Parameter Object Class code 0x93 or Host DPI Parameter Object Class code 0x9F for explicit messaging.

[4] In this example, we are writing to five parameters. Each parameter being written to requires an array of three INT registers. Therefore, a controller tag was created with its Data Type field set to “INT[15].” Also, the Source Length field on the Message Configuration screen must correspond to the selected Data Type in bytes (for this example, 30 bytes for an INT[15] array). Scattered write messages always assume that every parameter being written to is a 32-bit parameter, regardless of its actual size. Maximum length is 132 bytes or 66 words which equates to 22 parameters. For parameter numbering, see DPI Parameter Object on page C-16 (Class code 0x93) or Host DPI Parameter Object on page C-30 (Class code 0x9F).

ControlLogix Example Scattered Write Request Data

In this message example, we use the data structure in Figure 6.15 or Figure 6.18 in the source tag (Scattered_Write_Request) to write new values to these parameters:

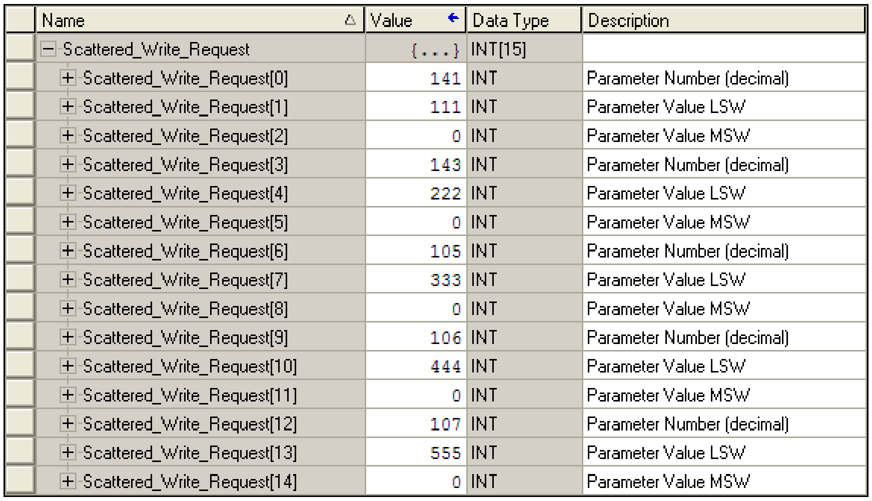
PowerFlex 7-Class Drive Parameter	Write Value
141 - [Accel Time 2]	11.1 Sec.
143 - [Decel Time 2]	22.2 Sec.
105 - [Preset Speed 5]	33.3 Hz.
106 - [Preset Speed 6]	44.4 Hz.
107 - [Preset Speed 7]	55.5 Hz.

PowerFlex 750-Series Drive Parameter	Write Value	Data Type
536 - [Accel Time 2]	11.1 Sec.	REAL
538 - [Decel Time 2]	22.2 Sec.	REAL
725 - [Zero Position]	33	DINT
21555 - [Port 5: Analog In0 Hi]	5.5	REAL
780 - [PTP Setpoint]	-75,555	REAL

Refer to DPI Parameter Object on page C-16 (Class code 0x93) or Host DPI Parameter Object on page C-30 (Class code 0x9F) for parameter numbering.

Figure 6.15 or Figure 6.18 shows the parameter values.

Figure 6.15 Example Scattered Write Request Data for PowerFlex 7-Class Drive

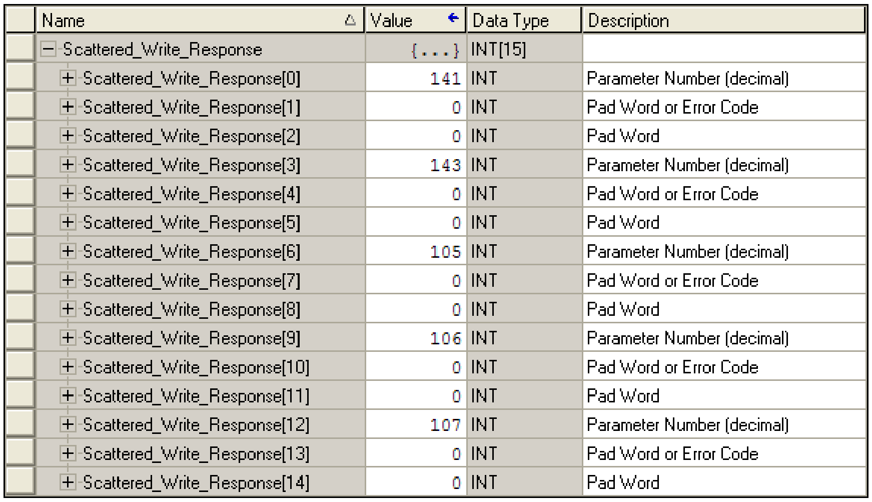


Name	Value	Data Type	Description
− Scattered_Write_Request	{...}	INT[15]	
+ Scattered_Write_Request[0]	141	INT	Parameter Number (decimal)
+ Scattered_Write_Request[1]	111	INT	Parameter Value LSW
+ Scattered_Write_Request[2]	0	INT	Parameter Value MSW
+ Scattered_Write_Request[3]	143	INT	Parameter Number (decimal)
+ Scattered_Write_Request[4]	222	INT	Parameter Value LSW
+ Scattered_Write_Request[5]	0	INT	Parameter Value MSW
+ Scattered_Write_Request[6]	105	INT	Parameter Number (decimal)
+ Scattered_Write_Request[7]	333	INT	Parameter Value LSW
+ Scattered_Write_Request[8]	0	INT	Parameter Value MSW
+ Scattered_Write_Request[9]	106	INT	Parameter Number (decimal)
+ Scattered_Write_Request[10]	444	INT	Parameter Value LSW
+ Scattered_Write_Request[11]	0	INT	Parameter Value MSW
+ Scattered_Write_Request[12]	107	INT	Parameter Number (decimal)
+ Scattered_Write_Request[13]	555	INT	Parameter Value LSW
+ Scattered_Write_Request[14]	0	INT	Parameter Value MSW

ControlLogix Example Scattered Write Response Data

The results of the message appear in the destination tag named Scattered_Write_Response (Figure 6.16). Values of “0” indicate no errors occurred.

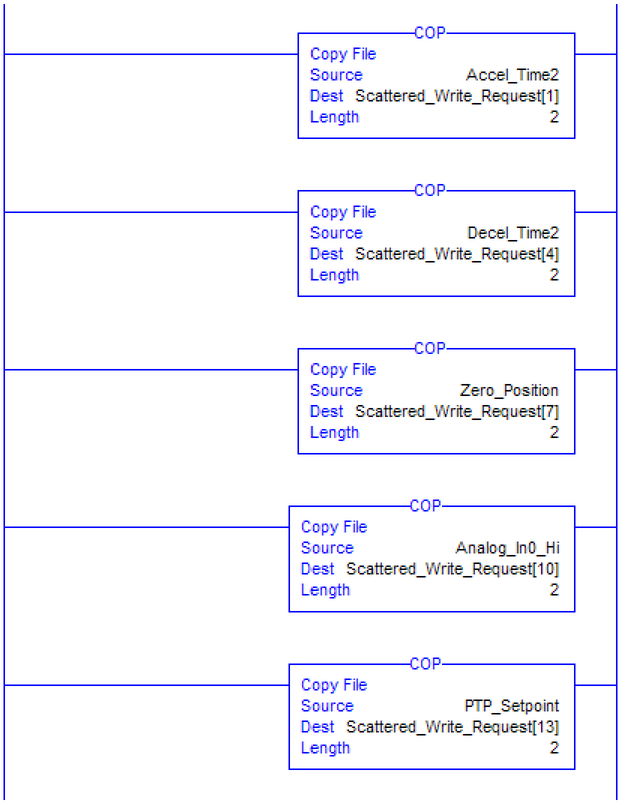
Figure 6.16 Example Scattered Write Response Data for PowerFlex 7-Class Drive



Name	Value	Data Type	Description
⊟ Scattered_Write_Response	{...}	INT[15]	
⊞ Scattered_Write_Response[0]	141	INT	Parameter Number (decimal)
⊞ Scattered_Write_Response[1]	0	INT	Pad Word or Error Code
⊞ Scattered_Write_Response[2]	0	INT	Pad Word
⊞ Scattered_Write_Response[3]	143	INT	Parameter Number (decimal)
⊞ Scattered_Write_Response[4]	0	INT	Pad Word or Error Code
⊞ Scattered_Write_Response[5]	0	INT	Pad Word
⊞ Scattered_Write_Response[6]	105	INT	Parameter Number (decimal)
⊞ Scattered_Write_Response[7]	0	INT	Pad Word or Error Code
⊞ Scattered_Write_Response[8]	0	INT	Pad Word
⊞ Scattered_Write_Response[9]	106	INT	Parameter Number (decimal)
⊞ Scattered_Write_Response[10]	0	INT	Pad Word or Error Code
⊞ Scattered_Write_Response[11]	0	INT	Pad Word
⊞ Scattered_Write_Response[12]	107	INT	Parameter Number (decimal)
⊞ Scattered_Write_Response[13]	0	INT	Pad Word or Error Code
⊞ Scattered_Write_Response[14]	0	INT	Pad Word

The PowerFlex 750-Series drive uses 32-bit integer and REAL parameters. A COP command must be used to copy the 32-bit integer and REAL values to the Scattered_Write_Request integer array. Figure 6.17 shows the ladder logic used for this example. If the parameter data type is a REAL, then the source tag is a REAL. If the parameter data type is a 32-bit integer, then the source tag is a DINT. See the drive documentation to determine the parameter data type (32-bit integer or REAL).

Figure 6.17 Example Ladder Logic to Copy Request Data for PowerFlex 750-Series Drive



COP
Copy File
Source Accel_Time2
Dest Scattered_Write_Request[1]
Length 2

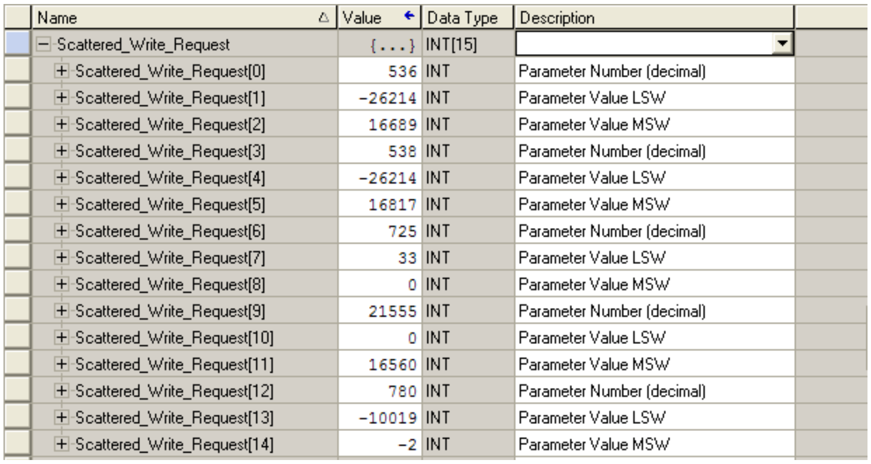
COP
Copy File
Source Decel_Time2
Dest Scattered_Write_Request[4]
Length 2

COP
Copy File
Source Zero_Position
Dest Scattered_Write_Request[7]
Length 2

COP
Copy File
Source Analog_In0_Hi
Dest Scattered_Write_Request[10]
Length 2

COP
Copy File
Source PTP_Setpoint
Dest Scattered_Write_Request[13]
Length 2

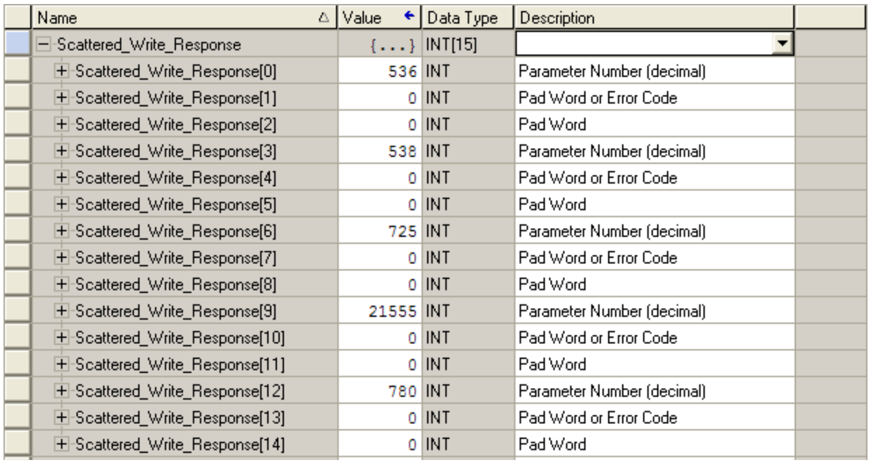
Figure 6.18 Example Scattered Write Request Data for PowerFlex 750-Series Drive



Name	Value	Data Type	Description
⊟ Scattered_Write_Request	{...}	INT[15]	
⊞ Scattered_Write_Request[0]	536	INT	Parameter Number (decimal)
⊞ Scattered_Write_Request[1]	-26214	INT	Parameter Value LSW
⊞ Scattered_Write_Request[2]	16689	INT	Parameter Value MSW
⊞ Scattered_Write_Request[3]	538	INT	Parameter Number (decimal)
⊞ Scattered_Write_Request[4]	-26214	INT	Parameter Value LSW
⊞ Scattered_Write_Request[5]	16817	INT	Parameter Value MSW
⊞ Scattered_Write_Request[6]	725	INT	Parameter Number (decimal)
⊞ Scattered_Write_Request[7]	33	INT	Parameter Value LSW
⊞ Scattered_Write_Request[8]	0	INT	Parameter Value MSW
⊞ Scattered_Write_Request[9]	21555	INT	Parameter Number (decimal)
⊞ Scattered_Write_Request[10]	0	INT	Parameter Value LSW
⊞ Scattered_Write_Request[11]	16560	INT	Parameter Value MSW
⊞ Scattered_Write_Request[12]	780	INT	Parameter Number (decimal)
⊞ Scattered_Write_Request[13]	-10019	INT	Parameter Value LSW
⊞ Scattered_Write_Request[14]	-2	INT	Parameter Value MSW

The results of the explicit message appear in the destination tag Scattered_Write_Response (Figure 6.19). Values of "0" indicate no errors occurred.

Figure 6.19 Example Scattered Write Response Data for PowerFlex 750-Series Drive



Name	Value	Data Type	Description
⊟ Scattered_Write_Response	{...}	INT[15]	
⊞ Scattered_Write_Response[0]	536	INT	Parameter Number (decimal)
⊞ Scattered_Write_Response[1]	0	INT	Pad Word or Error Code
⊞ Scattered_Write_Response[2]	0	INT	Pad Word
⊞ Scattered_Write_Response[3]	538	INT	Parameter Number (decimal)
⊞ Scattered_Write_Response[4]	0	INT	Pad Word or Error Code
⊞ Scattered_Write_Response[5]	0	INT	Pad Word
⊞ Scattered_Write_Response[6]	725	INT	Parameter Number (decimal)
⊞ Scattered_Write_Response[7]	0	INT	Pad Word or Error Code
⊞ Scattered_Write_Response[8]	0	INT	Pad Word
⊞ Scattered_Write_Response[9]	21555	INT	Parameter Number (decimal)
⊞ Scattered_Write_Response[10]	0	INT	Pad Word or Error Code
⊞ Scattered_Write_Response[11]	0	INT	Pad Word
⊞ Scattered_Write_Response[12]	780	INT	Parameter Number (decimal)
⊞ Scattered_Write_Response[13]	0	INT	Pad Word or Error Code
⊞ Scattered_Write_Response[14]	0	INT	Pad Word

ControlLogix – Explanation of Request and Response Data for Read/Write Multiple Messaging (all versions)

The data structures in Figure 6.20 and Figure 6.21 use 16-bit words and can accommodate up to 22 parameters in a single message. In the Response Message, a parameter number with the high bit set indicates that the associated parameter value field contains an error code (parameter number in response data will be negative).

Important: See Table 6.A on page 6-2 for limitations of PowerFlex 7-Class and PowerFlex 750-Series drives when using Class code 0x93 or Class code 0x9F for explicit messaging.

Figure 6.20 Data Structures for Scattered Read Messages

	Request (Source Data)		Response (Destination Data)
Word 0	Parameter Number	Word 0	Parameter Number
1	Pad Word	1	Parameter Value LSW
2	Pad Word	2	Parameter Value MSW
3	Parameter Number	3	Parameter Number
4	Pad Word	4	Parameter Value LSW
5	Pad Word	5	Parameter Value MSW
6	Parameter Number	6	Parameter Number
7	Pad Word	7	Parameter Value LSW
8	Pad Word	8	Parameter Value MSW
9	Parameter Number	9	Parameter Number
10	Pad Word	10	Parameter Value LSW
11	Pad Word	11	Parameter Value MSW
12	Parameter Number	12	Parameter Number
13	Pad Word	13	Parameter Value LSW
14	Pad Word	14	Parameter Value MSW
⋮		⋮	
63	Parameter Number	63	Parameter Number
64	Pad Word	64	Parameter Value LSW
65	Pad Word	65	Parameter Value MSW

Figure 6.21 Data Structures for Scattered Write Messages

	Request (Source Data)		Response (Destination Data)
Word 0	Parameter Number	Word 0	Parameter Number
1	Parameter Value LSW	1	Pad Word
2	Parameter Value MSW	2	Pad Word
3	Parameter Number	3	Parameter Number
4	Parameter Value LSW	4	Pad Word
5	Parameter Value MSW	5	Pad Word
6	Parameter Number	6	Parameter Number
7	Parameter Value LSW	7	Pad Word
8	Parameter Value MSW	8	Pad Word
9	Parameter Number	9	Parameter Number
10	Parameter Value LSW	10	Pad Word
11	Parameter Value MSW	11	Pad Word
12	Parameter Number	12	Parameter Number
13	Parameter Value LSW	13	Pad Word
14	Parameter Value MSW	14	Pad Word
⋮		⋮	
63	Parameter Number	63	Parameter Number
64	Parameter Value LSW	64	Pad Word
65	Parameter Value MSW	65	Pad Word

Appendix C

EtherNet/IP Objects

Appendix C presents information about the EtherNet/IP objects that can be accessed using Explicit Messages. For information on the format of Explicit Messages and example ladder logic programs, refer to Chapter 6, Using Explicit Messaging.

Object	Class Code		Page
	Hex.	Dec.	
Identity Object	0x01	1	C-2
Assembly Object	0x04	4	C-3
Register Object	0x07	7	C-4
Parameter Object [1]	0x0F	15	C-5
Parameter Group Object [1]	0x10	16	C-7
PCCC Object	0x67	103	C-8
DPI Device Object	0x92	146	C-13
DPI Parameter Object	0x93	147	C-16
DPI Fault Object	0x97	151	C-22
DPI Alarm Object	0x98	152	C-24
DPI Diagnostic Object	0x99	153	C-26
DPI Time Object	0x9B	155	C-28
Host DPI Parameter Object [2]	0x9F	159	C-30
TCP/IP Interface Object	0xF5	245	C-36
Ethernet Link Object	0xF6	246	C-37

[1] These objects are NOT supported when the adapter is used with a PowerFlex 750-Series drive.

[2] This object is supported only when the adapter is used with a PowerFlex 750-Series drive.

TIP: Refer to the EtherNet/IP specification for more information about EtherNet/IP objects. Information about the EtherNet/IP specification is available on the ODVA web site (http://www.odva.org).

Supported Data Types

Data Type	Description
BOOL	8-bit value -- low bit is true or false
BOOL[n]	Array of n bits
BYTE	8-bit unsigned integer
CONTAINER	32-bit parameter value - sign extended if necessary
DINT	32-bit signed integer
DWORD	32-bit unsigned integer
INT	16-bit signed integer
LWORD	64-bit unsigned integer
REAL	32-bit floating point
SHORT_STRING	1-byte length indicator + that many characters
SINT	8-bit signed integer
STRING[n]	Array of n characters
STRUCT	Structure name only - no size in addition to elements
TCHAR	8 or 16-bit character
UDINT	32-bit unsigned integer
UINT	16-bit unsigned integer
USINT	8-bit unsigned integer
WORD	16-bit unsigned integer

Identity Object

Class Code

Hexadecimal	Decimal
0x01	1

Services

	Implemented for:		
Service Code	Class	Instance	Service Name
0x05	No	Yes	Reset
0x0E	Yes	Yes	Get_Attribute_Single
0x01	Yes	Yes	Get_Attributes_All

Instances

The number of instances depends on the number of components in the device connected to the adapter. This number of components can be read in Instance 0, Attribute 2.

Instance	Description
0	Class
1	Host
2…15	Peripherals on Ports 1…14

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
2	Get	Max Instance	WORD	Total number of instances

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Vendor ID	WORD	1 = Allen-Bradley
2	Get	Device Type	WORD	123
3	Get	Product Code	WORD	Number identifying product name and rating
4	Get	Revision: Major Minor	STRUCT of: BYTE BYTE	 Value varies Value varies
5	Get	Status	WORD	Bit 0 = Owned Bit 8 = Minor recoverable fault Bit 10 = Major recoverable fault
6	Get	Serial Number	DWORD	Unique 32-bit number
7	Get	Product Name	SHORT_STRING	Product name and rating

Assembly Object

Class Code

Hexadecimal	Decimal
0x04	4

Services

	Implemented for:		
Service Code	Class	Instance	Service Name
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Instances

Instance	Description
1	All I/O data being read from the DPI device (read-only)
2	All I/O data written to the DPI device (read/write)

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Revision	WORD	2
2	Get	Max Instance	WORD	2
100	Set	Control Timeout	WORD	Control timeout in seconds

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Number of Members	WORD	1
2	Get	Member List	ARRAY of STRUCT: WORD WORD Packed EPATH	 Size of member data Size of member path Member path
3	Conditional [1]	Data	Array of Bits	Data to be transferred
4	Get	Size	WORD	Size of assembly data in bits

[1] For instance 1, access rule for the data attribute is Get. For instance 2, it is Get/Set.

Important: Setting an assembly object attribute can be done only when the Control Timeout (class attribute 100) has been set to a non-zero value.

Register Object

Class Code

Hexadecimal	Decimal
0x07	7

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Instances

Instance	Description
1	All I/O data being read from the DPI device (read-only)
2	All I/O data written to the DPI device (read/write)
3	Logic Status and Feedback data (read-only)
4	Logic Command and Reference data (read/write)
5	Datalink A (input data from device to scanner) (read only)
6	Datalink A (output data from scanner to device) (read/write)
7	Datalink B (input data from device to scanner) (read only)
8	Datalink B (output data from scanner to device) (read/write)
9	Datalink C (input data from device to scanner) (read only)
10	Datalink C (output data from scanner to device) (read/write)
11	Datalink D (input data from device to scanner) (read only)
12	Datalink D (output data from scanner to device) (read/write)
13	Logic Status and Feedback Data (read-only)
14	Mask [1] (read/write)
15	Logic Status (read-only)
16	Logic Command (read/write)
17	Feedback (read-only)
18	Reference (read/write)

[1] The mask command word is set to the value of the first word of the data where there are ones in the second word of the data. Command = (word 1 and not word 2) or (word 1 and word 2). This only controls specified bits in the Logic Command data to the DPI product and does not change the Reference value.

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
100	Set	Control Timeout	WORD	Control timeout in seconds

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Bad Flag	BOOL	If set to 1, then attribute 4 may contain invalid data. 0 = good 1 = bad
2	Get	Direction	BOOL	Direction of data transfer 0 = Producer Register (drive to network) 1 = Consumer Register (network to drive)
3	Get	Size	WORD	Size of register data in bits
4	Conditional [1]	Data	ARRAY of BITS	Data to be transferred

[1] For this attribute, the Access Rule is Get if Direction = 0. The Access Rule is Set if Direction = 1.

Important: Setting a Register object attribute can be done only when the Control Timeout (class attribute 100) has been set to a non-zero value.

Parameter Object

Class Code

Hexadecimal	Decimal
0x0F	15

Important: This object is not supported when the adapter is used with a PowerFlex 750-Series drive.

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x01	Yes	Yes	Get_Attributes_All
0x05	Yes	No	Reset
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single
0x4B	No	Yes	Get_Enum_String

Instances

The number of instances depends on the number of parameters in the DPI drive. The adapter parameters are appended to the list of drive parameters. The total number of parameters can be read in Instance 0, Attribute 2.

Instance	Description
0	Class Attributes
1	Drive Parameter 1 Attributes
⋮	⋮
n	Last Drive Parameter n Attributes [1]
n + 1	Adapter Parameter 1 Attributes
⋮	⋮
n + m	Last Adapter Parameter m Attributes [2]

[1] n represents the number of parameters in the drive.

[2] m represents the number of parameters in the adapter.

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Revision	WORD	1
2	Get	Max Instance	WORD	Number of parameters
8	Get	Parameter Class Descriptor	WORD	0 = False, 1 = True Bit 0 = Supports parameter instances Bit 1 = Supports full attributes Bit 2 = Must do NVS save command Bit 3 = Parameters are stored in NVS
9	Get	Configuration Assembly Instance	WORD	0
10	Set	Native Language	BYTE	0 = English 1 = French 2 = Spanish 3 = Italian 4 = German 5 = Japanese 6 = Portuguese 7 = Mandarin Chinese 8 = Russian 9 = Dutch

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	[1]	Parameter Value	[2]	[3]
2	Get	Link Path Size	BYTE	0 = No link specified n = The size of Attribute 3 in bytes
3	Get	Link Path		[4]
4	Get	Descriptor	WORD	0 = False, 1 = True Bit 1 = Supports ENUMs Bit 2 = Supports scaling Bit 3 = Supports scaling links Bit 4 = Read only Bit 5 = Monitor Bit 6 = Extended precision scaling
5	Get	Data Type	BYTE	1 = WORD (16-bit) 2 = UINT (16-bit) 3 = INT (16-bit) 5 = SINT 6 = DINT 8 = USINT 9 = UDINT 11 = REAL 22 = SHORT_STRING 24 = BYTE 25 = DWORD
6	Get	Data Size	BYTE	[3]
7	Get	Parameter Name String	SHORT_STRING	[3]
8	Get	Units String	SHORT_STRING	[3]
9	Get	Help String	SHORT_STRING	Null string
10	Get	Minimum Value	[2]	[3]
11	Get	Maximum Value	[2]	[3]
12	Get	Default Value	[2]	[3]
13	Get	Scaling Multiplier	WORD	[3]
14	Get	Scaling Divisor	WORD	[3]
15	Get	Scaling Base	WORD	[3]
16	Get	Scaling Offset	WORD	[3]
17	Get	Multiplier Link	WORD	[3]
18	Get	Divisor Link	WORD	[3]
19	Get	Base Link	WORD	[3]
20	Get	Offset Link	WORD	[3]
21	Get	Decimal Precision	BYTE	[3]

[1] Access rule is defined in Bit 4 of instance attribute 4 (0 = Get/Set, 1 = Get).

[2] Specified in descriptor, data type, and data size.

[3] Value varies based on parameter instance.

[4] Refer to the EtherNet/IP specification for a description of the link path.

Parameter Group Object

Class Code

Hexadecimal	Decimal
0x10	16

Important: This object is not supported when the adapter is used with a PowerFlex 750-Series drive.

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	No	Set_Attribute_Single

Instances

The number of instances depends on the number of groups in the device. A group of adapter parameters is appended to the list of groups in the device. The total number of groups can be read in Instance 0, Attribute 2.

Number	Description
0	Class Attributes
1	Drive Group 1 Attributes
⋮	⋮
n	Last Drive Group Attributes
n + 1	Adapter Group Attributes

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Parameter group version	WORD	1
2	Get	Max Instance	WORD	Total number of groups
8	Set	Native Language	BYTE	0 = English 1 = French 2 = Spanish 3 = Italian 4 = German 5 = Japanese 6 = Portuguese 7 = Mandarin Chinese 8 = Russian 9 = Dutch

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Group Name String	SHORT_STRING	Group name
2	Get	Number of Members in Group	WORD	Number of parameters in group
3	Get	1st Parameter Number in Group	WORD	[1]
4	Get	2nd Parameter Number in Group	WORD	[1]
n	Get	⋮	WORD	[1]

[1] Value varies based on group instance.

PCCC Object

Class Code

Hexadecimal	Decimal
0x67	103

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x4B	No	Yes	Execute_PCCC
0x4C	No	Yes	Execute_DH+

Instances

Supports Instance 1.

Class Attributes

Not supported.

Instance Attributes

Not supported.

Message Structure for Execute_PCCC

Request		
Name	Data Type	Description
Length	USINT	Length of requestor ID
Vendor	UINT	Vendor number of requestor
Serial Number	UDINT	ASA serial number of requestor
Other	Product Specific	Identifier of user, task, etc. on the requestor
CMD	USINT	Command byte
STS	USINT	0
TNSW	UINT	Transport word
FNC	USINT	Function code; not used for all CMDs.
PCCC_ params	ARRAY of USINT	CMD/FNC specific parameters

Response		
Name	Data Type	Description
Length	USINT	Length of requestor ID
Vendor	UINT	Vendor number of requestor
Serial Number	UDINT	ASA serial number of requestor
Other	Product Specific	Identifier of user, task, etc. on the requestor
CMD	USINT	Command byte
STS	USINT	Status byte
TNSW	UINT	Transport word. Same value as the request.
EXT_STS	USINT	Extended status; not used for all CMDs.
PCCC_ results	ARRAY of USINT	CMD/FNC specific result data

Message Structure for Execute_DH+

Request		
Name	**Data Type**	**Description**
DLink	UINT	Destination Link ID
DSta	USINT	Destination Station number
DUser	USINT	Destination “User” number
SLink	UINT	Source Link ID
SSta	USINT	Source Station number
SUser	USINT	Source User number
CMD	USINT	Command byte
STS	USINT	0
TNSW	UINT	Transport word
FNC	USINT	Function code; not used for all CMDs
PCCC_ params	ARRAY of USINT	CMD/FNC specific parameters

Response		
Name	**Data Type**	**Description**
DLink	UINT	Destination Link ID
DSta	USINT	Destination Station number
DUser	USINT	Destination “User” number
SLink	UINT	Source Link ID
SSta	USINT	Source Station number
SUser	USINT	Source User number
CMD	USINT	Command byte
STS	USINT	Status byte
TNSW	UINT	Transport word. Same value as the request.
EXT_STS	USINT	Extended status; not used for all CMDs
PCCC_ results	ARRAY of USINT	CMD/FNC specific result data

The adapter supports the following PCCC command types:

CMD	FNC	Description
0x06	0x03	Identify host and some status
0x0F	0x67	PLC-5 typed write
0x0F	0x68	PLC-5 typed read
0x0F	0x95	Encapsulate other protocol
0x0F	0xA2	SLC 500 protected typed read with 3 address fields
0x0F	0xAA	SLC 500 protected typed write with 3 address fields
0x0F	0xA1	SLC 500 protected typed read with 2 address fields
0x0F	0xA9	SLC 500 protected typed write with 2 address fields
0x0F	0x00	Word range read
0x0F	0x01	Word range write

For more information regarding PCCC commands, see *DF1 Protocol and Command Set Manual* (Allen-Bradley publication 1770-6.5.16).

N-Files

N-File	Description			
N40	This N-file lets you use Emulated Block Transfer messages to read and write many types of DPI messages. To use Emulated Block Transfer messages, you send a Write message to N40:0…N40:63, wait until the adapter responds with a reply message, and then read the response data in N40:0…N40:63 with a Read message. For details about Block Transfer messages and the data required for each byte in the N-File, refer to the *Remote I/O Adapter User Manual* (publication 20COMM-UM004). Bits 15…8 are the Most Significant Byte. Bits 7…0 are the Least Significant Byte.			
	Write		*Read*	
Bits	15	0	15	0
N40:0	0x00	Length (in Bytes)	0x00	Length (in Bytes)
N40:1	DPI Port	0x81	Status Size	Status Type
N40:2	0x00	CIP Service	Data (length varies based on message)	
N40:3	CIP Class			
N40:4	CIP Instance			
N40:5	CIP Attribute			
N40:6 ⋮ N40:63	Data (length varies based on message)			
N41	This N-file lets you read and write control I/O messages. You can write control I/O messages only when all of the following conditions are true: • The adapter is not receiving I/O from a scanner. For example, there is no scanner on the network, the scanner is in idle (program) mode, the scanner is faulted, or the adapter is not mapped to the scanner. • The adapter is not receiving Peer I/O from another adapter. • The value of N42:3 is set to a non-zero value.			
	Write		*Read*	
N41:0	Logic Command Word		Logic Status Word	
N41:1	Reference (least significant word)		Feedback (least significant word)	
N41:2	Reference (most significant word)		Feedback (most significant word)	
N41:3	Datalink A1 (least significant word)		Datalink A1 (least significant word)	
N41:4	Datalink A1 (most significant word)		Datalink A1 (most significant word)	
N41:5	Datalink A2 (least significant word)		Datalink A2 (least significant word)	
N41:6	Datalink A2 (most significant word)		Datalink A2 (most significant word)	
N41:7	Datalink B1 (least significant word)		Datalink B1 (least significant word)	
N41:8	Datalink B1 (most significant word)		Datalink B1 (most significant word)	
N41:9	Datalink B2 (least significant word)		Datalink B2 (least significant word)	
N41:10	Datalink B2 (most significant word)		Datalink B2 (most significant word)	
N41:11	Datalink C1 (least significant word)		Datalink C1 (least significant word)	
N41:12	Datalink C1 (most significant word)		Datalink C1 (most significant word)	
N41:13	Datalink C2 (least significant word)		Datalink C2 (least significant word)	
N41:14	Datalink C2 (most significant word)		Datalink C2 (most significant word)	
N41:15	Datalink D1 (least significant word)		Datalink D1 (least significant word)	
N41:16	Datalink D1 (most significant word)		Datalink D1 (most significant word)	
N41:17	Datalink D2 (least significant word)		Datalink D2 (least significant word)	
N41:18	Datalink D2 (most significant word)		Datalink D2 (most significant word)	
N42	This N-file lets you read and write some values configuring the port.			
N42:3	Time-out (read/write): Time (in seconds) allowed between messages to the N41 file. If the adapter does not receive a message in the specified time, it performs the fault action configured in its [Comm Flt Action] parameter. A valid setting is between 1 and 32767 seconds (5…20 seconds is recommended).			
N42:7	Adapter Port Number (read only): DPI port on the drive to which the adapter is connected.			
N42:8	Peer Adapters (read only): Bit field of devices having DPI Peer capabilities.			

N45	This N-file lets you read and write control I/O messages. You can write control I/O messages only when all of the following conditions are true: • The adapter is not receiving I/O from a scanner. For example, there is no scanner on the network, the scanner is in idle (program) mode, the scanner is faulted, or the adapter is not mapped to the scanner. • The adapter is not receiving Peer I/O from another adapter. • The value of N42:3 is set to a non-zero value.	
	Write	*Read*
N45:0	Logic Command (least significant)	Logic Status (least significant)
N45:1	Logic Command (most significant)	Logic Status (most significant)
N45:2	Reference (least significant)	Feedback (least significant)
N45:3	Reference (most significant)	Feedback (most significant)
N45:4	DL From Net 01 (least significant)	DL To Net 01 (least significant)
N45:5	DL From Net 01 (most significant)	DL To Net 01 (most significant)
N45:6	DL From Net 02 (least significant)	DL To Net 02 (least significant)
N45:7	DL From Net 02 (most significant)	DL To Net 02 (most significant)
N45:8	DL From Net 03 (least significant)	DL To Net 03 (least significant)
N45:9	DL From Net 03 (most significant)	DL To Net 03 (most significant)
N45:10	DL From Net 04 (least significant)	DL To Net 04 (least significant)
N45:11	DL From Net 04 (most significant)	DL To Net 04 (most significant)
N45:12	DL From Net 05 (least significant)	DL To Net 05 (least significant)
N45:13	DL From Net 05 (most significant)	DL To Net 05 (most significant)
N45:14	DL From Net 06 (least significant)	DL To Net 06 (least significant)
N45:15	DL From Net 06 (most significant)	DL To Net 06 (most significant)
N45:16	DL From Net 07 (least significant)	DL To Net 07 (least significant)
N45:17	DL From Net 07 (most significant)	DL To Net 07 (most significant)
N45:18	DL From Net 08 (least significant)	DL To Net 08 (least significant)
N45:19	DL From Net 08 (most significant)	DL To Net 08 (most significant)
N45:20	DL From Net 09 (least significant)	DL To Net 09 (least significant)
N45:21	DL From Net 09 (most significant)	DL To Net 09 (most significant)
N45:22	DL From Net 10 (least significant)	DL To Net 10 (least significant)
N45:23	DL From Net 10 (most significant)	DL To Net 10 (most significant)
N45:24	DL From Net 11 (least significant)	DL To Net 11 (least significant)
N45:25	DL From Net 11 (most significant)	DL To Net 11 (most significant)
N45:26	DL From Net 12 (least significant)	DL To Net 12 (least significant)
N45:27	DL From Net 12 (most significant)	DL To Net 12 (most significant)
N45:28	DL From Net 13 (least significant)	DL To Net 13 (least significant)
N45:29	DL From Net 13 (most significant)	DL To Net 13 (most significant)
N45:30	DL From Net 14 (least significant)	DL To Net 14 (least significant)
N45:31	DL From Net 14 (most significant)	DL To Net 14 (most significant)
N45:32	DL From Net 15 (least significant)	DL To Net 15 (least significant)
N45:33	DL From Net 15 (most significant)	DL To Net 15 (most significant)
N45:34	DL From Net 16 (least significant)	DL To Net 16 (least significant)
N45:35	DL From Net 16 (most significant)	DL To Net 16 (most significant)

Important: If your controller or HMI platform supports CIP messaging, use the CIP Parameter object to get and set parameters.

N-File	Description
N150…N199	These N-files let you read and write parameter values in DPI Port 0 (the host PowerFlex drive) as 32-bit double words. You can interpret the data in various ways (for example, 32-bit real, 32-bit integer) To read a parameter, you need to send a message with two elements. For example, to read parameter 1, read two elements beginning at N150:2. As another example, to read parameters 2…6, read ten elements beginning at N150:4.
N150:0…1	Number of parameters in the drive
N150:2…249	Drive parameters 1…124
N151:0…249	Drive parameters 125…249
N152:0…249	Drive parameters 250…374
N153:0…249	Drive parameters 375…499
⋮	⋮
N199:0…249	Drive parameters 6125…6249

N-File	Description
N201…N212	These N-files let you read and write values to DPI and Host parameters in Ports 1…14 (for example, a HIM or adapter) as 32-bit double words. You can interpret the data in various ways (for example, 32-bit real, 32-bit integer) To read a parameter, you need to send a message with two elements. For example, to read parameter 1 in the peripheral connected to DPI port 1, read two elements beginning at N201:2. As another example, to read parameters 2…6 in the peripheral connected to DPI port 5 (the adapter), read ten elements beginning at N209:4.
N201:0…1	Number of parameters in the DPI peripheral at DPI port 1
N201:2…249	Parameters 1…124 in the DPI peripheral at DPI port 1
N202:0…249	Parameters 125…249 in the DPI peripheral at DPI port 1
N203:0…1	Number of parameters in the DPI peripheral at DPI port 2
N203:2…249	Parameters 1…124 in the DPI peripheral at DPI port 2
N204:0…249	Parameters 125…249 in the DPI peripheral at DPI port 2
N205:0…1	Number of parameters in the DPI peripheral at DPI port 3
N205:2…249	Parameters 1…124 in the DPI peripheral at DPI port 3
N206:0…249	Parameters 125…249 in the DPI peripheral at DPI port 3
N207:0…1	Number of parameters in the DPI peripheral at DPI port 4
N207:2…249	Parameters 1…124 in the DPI peripheral at DPI port 4
N208:0…249	Parameters 125…249 in the DPI peripheral at DPI port 4
N209:0…1	Number of parameters in the DPI peripheral at DPI port 5
N209:2…249	Parameters 1…124 in the DPI peripheral at DPI port 5
N210:0…249	Parameters 125…249 in the DPI peripheral at DPI port 5
N211:0…1	Number of parameters in the DPI peripheral at DPI port 6
N211:2…249	Parameters 1…124 in the DPI peripheral at DPI port 6
N212:0…249	Parameters 125…249 in the DPI peripheral at DPI port 6
The following N-Files are supported only when the adapter is used with a PowerFlex 750-Series drive.	
N213:0…1	Number of parameters in the DPI peripheral at DPI port 7
N213:2…249	Parameters 1…124 in the DPI peripheral at DPI port 7
N214:0…249	Parameters 125…249 in the DPI peripheral at DPI port 7
N215:0…1	Number of parameters in the DPI peripheral at DPI port 8
N215:2…249	Parameters 1…124 in the DPI peripheral at DPI port 8
N216:0…249	Parameters 125…249 in the DPI peripheral at DPI port 8
N217:0…1	Number of parameters in the DPI peripheral at DPI port 9
N217:2…249	Parameters 1…124 in the DPI peripheral at DPI port 9
N218:0…249	Parameters 125…249 in the DPI peripheral at DPI port 9
N219:0…1	Number of parameters in the DPI peripheral at DPI port 10
N219:2…249	Parameters 1…124 in the DPI peripheral at DPI port 10
N220:0…249	Parameters 125…249 in the DPI peripheral at DPI port 10
N221:0…1	Number of parameters in the DPI peripheral at DPI port 11
N221:2…249	Parameters 1…124 in the DPI peripheral at DPI port 11
N222:0…249	Parameters 125…249 in the DPI peripheral at DPI port 11
N223:0…1	Number of parameters in the DPI peripheral at DPI port 12
N223:2…249	Parameters 1…124 in the DPI peripheral at DPI port 12
N224:0…249	Parameters 125…249 in the DPI peripheral at DPI port 12
N225:0…1	Number of parameters in the DPI peripheral at DPI port 13
N225:2…249	Parameters 1…124 in the DPI peripheral at DPI port 13
N226:0…249	Parameters 125…249 in the DPI peripheral at DPI port 13
N227:0…1	Number of parameters in the DPI peripheral at DPI port 14
N227:2…249	Parameters 1…124 in the DPI peripheral at DPI port 14
N228:0…249	Parameters 125…249 in the DPI peripheral at DPI port 14

DPI Device Object

Class Code

Hexadecimal	Decimal
0x92	146

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Instances

The number of instances depends on the number of components in the device. The total number of components can be read in Instance 0, Class Attribute 4.

Instances (Hex.)	(Dec.)	Device
0x0000…0x3FFF	0…16383	Host
0x4000…0x43FF	16384…17407	Adapter
0x4400…0x47FF	17408…18431	DPI Port 1
0x4800…0x4BFF	18432…19455	DPI Port 2
0x4C00…0x4FFF	19456…20479	DPI Port 3
0x5000…0x53FF	20480…21503	DPI Port 4
0x5400…0x57FF	21504…22527	DPI Port 5
0x5800…0x5BFF	22528…23551	DPI Port 6
0x5C00…0x5FFF [1]	22552…24575	DPI Port 7
0x6000…0x63FF [1]	24576…25599	DPI Port 8
0x6400…0x67FF [1]	25600…26623	DPI Port 9
0x6800…0x6BFF [1]	26624…27647	DPI Port 10
0x6C00…0x6FFF [1]	27648…28671	DPI Port 11
0x7000…0x73FF [1]	28672…29695	DPI Port 12
0x7400…0x77FF [1]	29696…30719	DPI Port 13
0x7800…0x7BFF [1]	30720…31743	DPI Port 14

Example	Description
0	Class Attributes (Drive)
1	Drive Component 1
2	Drive Component 2
⋮	⋮
16384	Class Attributes (Adapter)
16385	Adapter Component 1
⋮	⋮

[1] These instances are supported only when the adapter is used with a PowerFlex 750-Series drive.

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Get	Family Code	BYTE	0x00 = DPI Peripheral 0x30 = PowerFlex 70 0x34 = PowerFlex 700H 0x38, 0x39, or 0x3A= PowerFlex 700 0x40 = PowerFlex 7000 0x48, 0x49, or 0x4A = PowerFlex 700S 0x5A = SMC Flex 0x68, 0x69, or 0x6A = PowerFlex 700VC 0x90 = PowerFlex 753/755 0xA0 = 20-750-xxx Option Module 0xFF = HIM
1	Get	Family Text	STRING[16]	Text identifying the device.

Attribute ID	Access Rule	Name	Data Type	Description
2	Set	Language Code	BYTE	0 = English 1 = French 2 = Spanish 3 = Italian 4 = German 5 = Japanese 6 = Portuguese 7 = Mandarin Chinese 8 = Russian 9 = Dutch
3	Get	Product Series	BYTE	1 = A 2 = B …
4	Get	Number of Components	BYTE	Number of components (e.g., main control board, I/O boards) in the device.
5	Set	User Definable Text	STRING[16]	Text identifying the device with a user-supplied name.
6	Get	Status Text	STRING[12]	Text describing the status of the device.
7	Get	Configuration Code	BYTE	Identification of variations.
8	Get	Configuration Text	STRING[16]	Text identifying a variation of a family device.
9	Get	Brand Code	WORD	0x0001 = Allen-Bradley
11	Get	NVS Checksum	WORD	Checksum of the Non-Volatile Storage in a device.
12	Get	Class Revision	WORD	2 = DPI
13	Get	Character Set Code	BYTE	0 = SCANport HIM 1 = ISO 8859-1 (Latin 1) 2 = ISO 8859-2 (Latin 2) 3 = ISO 8859-3 (Latin 3) 4 = ISO 8859-4 (Latin 4) 5 = ISO 8859-5 (Cyrillic) 6 = ISO 8859-6 (Arabic) 7 = ISO 8859-7 (Greek) 8 = ISO 8859-8 (Hebrew) 9 = ISO 8859-9 (Turkish) 10 = ISO 8859-10 (Nordic) 255 = ISO 10646 (Unicode)
14	Get	Product Option Support	BOOL[64]	
15	Get	Languages Supported	STRUCT of: BYTE BYTE[n]	 Number of Languages Language Codes (see Class Attribute 2)
16	Get	Date of Manufacture	STRUCT of: WORD BYTE BYTE	 Year Month Day
17	Get	Product Revision	STRUCT of: BYTE BYTE	 Major Firmware Release Minor Firmware Release
18	Get	Serial Number	DWORD	Value between 0x00000000 and 0xFFFFFFFF
19	Set	Language Selected	BYTE	0 = Default (HIM will prompt at start up) 1 = Language was selected (no prompt)
20	Set	Customer-Generated Firmware	STRING[36]	GUID (Globally Unique Identifier) identifying customer firmware flashed into the device.
30	Get	International Status Text	STRINGN	Text describing the status of device with support for Unicode.
31	Get/Set	International User Definable Text	STRINGN	Text identifying the device with a user-supplied name with support for Unicode.

Attribute ID	Access Rule	Name	Data Type	Description
34	Get	Key Information	STRUCT of: DWORD DWORD WORD WORD WORD BYTE BYTE BYTE BYTE BYTE BYTE[16]	 Rating Code Device Serial Number Customization Code Customization Revision Brand Code Family Code Config Code Language Code Major Revision Minor Revision Customer-Generated Firmware UUID
35	Get	NVS CRC	DWORD	A 32-bit CRC of the Non-Volatile Storage in a device.
39	Get	SI Driver Code	WORD	Code identifying the protocol between the device and host.
128	Get	Customization Code	WORD	Code identifying the customized device.
129	Get	Customization Revision Number	WORD	Revision of the customized device.
130	Get	Customization Device Text	STRING[32]	Text identifying the customized device.

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
3	Get	Component Name	STRING[32]	Name of the component
4	Get	Component Firmware Revision	STRUCT of: BYTE BYTE	 Major Revision Minor Revision
8	Get	Component Serial Number	DWORD	Value between 0x00000000 and 0xFFFFFFFF
9	Get	International Component Name	STRINGN	Name of the component with support for Unicode.

DPI Parameter Object

Class Code

Hexadecimal	Decimal
0x93	147

To access “Host Config” parameters, use the HOST DPI Parameter Object (Class code 0x9F).

Instances

The number of instances depends on the number of parameters in the device. The total number of parameters can be read in Instance 0, Attribute 0.

Instances (Hex.)	(Dec.)	Device
0x0000…0x3FFF	0…16383	Host
0x4000…0x43FF	16384…17407	Adapter
0x4400…0x47FF	17408…18431	DPI Port 1
0x4800…0x4BFF	18432…19455	DPI Port 2
0x4C00…0x4FFF	19456…20479	DPI Port 3
0x5000…0x53FF	20480…21503	DPI Port 4
0x5400…0x57FF	21504…22527	DPI Port 5
0x5800…0x5BFF	22528…23551	DPI Port 6
0x5C00…0x5FFF [1]	23552…24575	DPI Port 7
0x6000…0x63FF [1]	24576…25599	DPI Port 8
0x6400…0x67FF [1]	25600…26623	DPI Port 9
0x6800…0x6BFF [1]	26624…27647	DPI Port 10
0x6C00…0x6FFF [1]	27648…28671	DPI Port 11
0x7000…0x73FF [1]	28672…29695	DPI Port 12
0x7400…0x77FF [1]	29696…30719	DPI Port 13
0x7800…0x7BFF [1]	30720…31743	DPI Port 14

[1] These instances are supported only when the adapter is used with a PowerFlex 750-Series drive.

Example	Description
0	Class Attributes (Drive)
1	Drive Parameter 1 Attributes
2	Drive Parameter 2 Attributes
⋮	⋮
16384	Class Attributes (Adapter)
16385	Adapter Parameter 1 Attributes
⋮	⋮

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Get	Number of Instances	WORD	Number of parameters in the device
1	Set	Write Protect Password	WORD	0 = Password disabled n = Password
2	Set	NVS Command Write	BYTE	0 = No Operation 1 = Store values in active memory to NVS 2 = Load values in NVS to active memory 3 = Load default values to active memory
3	Get	NVS Parameter Value Checksum	WORD	Checksum of all parameter values in a user set in NVS
4	Get	NVS Link Value Checksum	WORD	Checksum of parameter links in a user set in NVS
5	Get	First Accessible Parameter	WORD	First parameter available if parameters are protected by passwords. A “0” indicates all parameters are protected.
7	Get	Class Revision	WORD	2 = DPI
8	Get	First Parameter Processing Error	WORD	The first parameter that has been written with a value outside of its range. A “0” indicates no errors.
9	Set	Link Command	BYTE	0 = No Operation 1 = Clear All Parameter Links (This does not clear links to function blocks.)

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
6	Get	DPI Offline Read Full	STRUCT of: BOOL[32] CONTAINER CONTAINER CONTAINER STRING[16] STRING[4] WORD WORD WORD WORD WORD WORD WORD BYTE BYTE WORD WORD CONTAINER WORD WORD WORD INT	 Descriptor Offline minimum value Offline maximum value Offline default value Parameter name Offline parameter units Online minimum parameter instance Online maximum parameter instance Online default parameter instance Multiple parameter instance Divisor parameter instance Base parameter instance Offset parameter instance Formula number Pad byte (always zero) Help instance Pad word (always a value of zero) Parameter value Multiplier Divisor Base Offset
7	Get	DPI Online Read Full	STRUCT of: BOOL[32] CONTAINER[1] CONTAINER CONTAINER CONTAINER WORD WORD STRING[4] WORD WORD WORD INT BYTE[3] BYTE STRING[16]	 Descriptor (see page C-19) Parameter value Minimum value Maximum value Default value Next parameter Previous parameter Units (for example, Amp, Hz) Multiplier [2] Divisor [2] Base [2] Offset [2] Link (source of the value) (0 = no link) Always zero (0) Parameter name
8	Get	DPI Descriptor	BOOL[32]	Descriptor (see page C-19)
9	Get/Set	DPI Parameter Value	Various	Parameter value in NVS. [3]
10	Get/Set	DPI RAM Parameter Value	Various	Parameter value in temporary memory. Valid only for DPI drives.
11	Get/Set	DPI Link	BYTE[3]	Link (parameter or function block that is the source of the value) (0 = no link)
12	Get	Help Object Instance	WORD	ID for help text for this parameter
13	Get	DPI Read Basic	STRUCT of: BOOL[32] CONTAINER CONTAINER CONTAINER CONTAINER STRING[16] STRING[4]	 Descriptor (see page C-19) Parameter value Minimum value Maximum value Default value Parameter name Units (for example, Amp, Hz)
14	Get	DPI Parameter Name	STRING[16]	Parameter name
15	Get	DPI Parameter Alias	STRING[16]	Customer supplied parameter name.
16	Get	Parameter Processing Error	BYTE	0 = No error 1 = Value is less than the minimum 2 = Value is greater than the maximum

Attribute ID	Access Rule	Name	Data Type	Description
18	Get	International DPI Offline Parameter Text	STRUCT of:	
			STRINGN	International parameter name
			STRINGN	International offline units
19	Get	International DPI Online Parameter Text	STRUCT of:	
			STRINGN	International parameter name
			STRINGN	International offline units
20	Get	International DPI Online Read Full	STRUCT of:	
			BOOL[32]	Descriptor
			CONTAINER	Parameter value
			CONTAINER	Online minimum value
			CONTAINER	Online maximum value
			CONTAINER	Online default value
			WORD	Next
			WORD	Previous
			WORD	Multiplier
			WORD	Divisor
			WORD	Base
			INT	Offset
			BYTE[3]	Link
			BYTE	Pad word (always zero)
			BOOL[32]	Extended descriptor
			STRINGN	International parameter name
			STRINGN	International online parameter units
21	Get	DPI Extended Descriptor	DWORD	Extended descriptor (see page C-20)
22	Get	International DPI Offline Read Full	STRUCT of:	
			BOOL	Descriptor
			CONTAINER	Offline minimum value
			CONTAINER	Offline maximum value
			CONTAINER	Offline default value
			WORD	Online minimum parameter instance
			WORD	Online maximum parameter instance
			WORD	Online default parameter instance
			WORD	Multiple parameter instance
			WORD	Divisor parameter instance
			WORD	Base parameter instance
			WORD	Offset parameter instance
			BYTE	Formula number
			BYTE	Pad byte (always zero)
			WORD	Help instance
			WORD	Pad word (always a value of zero)
			CONTAINER	Parameter value
			WORD	Multiplier
			WORD	Divisor
			WORD	Base
			INT	Offset
			BOOL[32]	Extended DPI descriptor
			STRINGN	International DPI parameter name
			STRINGN	International DPI offline parameter units

[1] A CONTAINER is a 32-bit block of data that contains the data type used by a parameter value. If signed, the value is sign extended. Padding is used in the CONTAINER to ensure that it is always 32-bits.

[2] This value is used in the formulas used to convert the parameter value between display units and internal units. Refer to Formulas for Converting on page C-21.

[3] Do NOT continually write parameter data to NVS. Refer to the attention on page 6-1.

Descriptor Attributes

Bit	Name	Description
0	Data Type (Bit 1)	Right bit is least significant bit (0). 000 = BYTE used as an array of Boolean 001 = WORD used as an array of Boolean 010 = BYTE (8-bit integer) 011 = WORD (16-bit integer) 100 = DWORD (32-bit integer) 101 = TCHAR (8-bit (not unicode) or 16-bits (unicode) 110 = REAL (32-bit floating point value) 111 = Use bits 16, 17, 18
1	Data Type (Bit 2)	
2	Data Type (Bit 3)	
3	Sign Type	0 = Unsigned 1 = Signed
4	Hidden	0 = Visible 1 = Hidden
5	Not a Link Sink	0 = Parameter can sink a link 1 = Parameter cannot sink a link
6	Not Recallable	0 = Recallable from NVS 1 = Not Recallable from NVS
7	ENUM	0 = No ENUM text 1 = ENUM text
8	Writable	0 = Read only 1 = Read/write
9	Not Writable When Enabled	0 = Writable when enabled (for example, drive running) 1 = Not writable when enabled
10	Instance	0 = Parameter value is not a Reference to another parameter 1 = Parameter value refers to another parameter
11	Uses Bit ENUM Mask	This parameter instance supports the Bit ENUM Mask attribute. For more information, see the definition of the attribute.
12	Decimal Place (Bit 0)	Number of digits to the right of the decimal point. 0000 = 0 1111 = 15
13	Decimal Place (Bit 1)	
14	Decimal Place (Bit 2)	
15	Decimal Place (Bit 3)	
16	Extended Data Type (Bit 4)	Right bit is least significant bit (16). 000 = Reserved 001 = DWORD used as an array of Boolean 010 = Reserved 011 = Reserved 100 = Reserved 101 = Reserved 110 = Reserved 111 = Reserved
17	Extended Data Type (Bit 5)	
18	Extended Data Type (Bit 6)	
19	Parameter Exists	Used to mark parameters that are not available to network tools.
20	Not Used	Reserved
21	Formula Links	Indicates the Formula Data is derived from other parameters.
22	Access Level (Bit 1)	A 3-bit field used to control access to parameter data.
23	Access Level (Bit 2)	
24	Access Level (Bit 3)	
25	Writable ENUM	ENUM text: 0 = Read Only, 1 = Read/Write
26	Not a Link Source	0 = May be the source end of a link 1 = May not be the source end of a link
27	Enhanced Bit ENUM	Parameter supports enhanced bit ENUMs.
28	Enhanced ENUM	Parameter supports enhanced ENUMs.
29	Uses DPI Limits Object	Parameter uses the DPI LImits Object.
30	Extended Descriptor	Parameter uses Extended Descriptor bits, which can be obtained by reading the DPI Extended Descriptor attribute for this parameter.
31	Always Upload/Download	Parameter shall always be included in uploads and downloads.

Extended Descriptor Attributes

Bit	Name	Description
0	Indirect Mode	0 = Analog (selects entire parameters) 1 = Digital (selects individual bits within parameters)
1	Indirect Type 0	Analog input list (Instance 0xFFFF)
2	Indirect Type 1	Digital input list (Instance 0xFFFE)
3	Indirect Type 2	Feedback list (Instance 0xFFFD)
4	Indirect Type 3	Analog output list (Instance 0xFFFC)
5	Indirect Type 4	Digital output list (Instance 0xFFFB)
6	Indirect Type 5	Undefined (Instance 0xFFFA)
7	Indirect Type 6	Undefined (Instance 0xFFF9)
8	Indirect Type 7	Undefined (Instance 0xFFF8)
9	Indirect Type 8	Undefined (Instance 0xFFF7)
10	Indirect Type 9	Undefined (Instance 0xFFF6)
11	Indirect Type 10	Undefined (Instance 0xFFF5)
12	Indirect Type 11	Undefined (Instance 0xFFF4)
13	Indirect Type 12	Undefined (Instance 0xFFF3)
14	Indirect Type 13	Undefined (Instance 0xFFF2)
15	Indirect Type 14	Parameter-specific list
16	FP Max Decimals Bit 0	These four bits are used on REAL parameters only. They indicate the maximum number of decimal places to be displayed for small values. A value of 0 indicates to not limit the number of decimal places used.
17	FP Max Decimals Bit 1	
18	FP Max Decimals Bit 2	
19	FP Max Decimals Bit 1	
20	Extended Parameter Reference	0 = Not an Extended Parameter Reference 1 = Extended Parameter Reference An Extended Parameter Reference contains a reference to another parameter. The value is formatted the same as an analog mode Indirect Selector parameter (SSpppp, where SS = slot number of device to which this Extended Parameter Reference is pointing, and pppp = number of the parameter or diagnostic item to which this Extended Parameter Reference is pointing). Note that an Extended Parameter Reference can only select parameters unlike an Indirect Selector. An Extended Parameter Reference could be used to configure a Datalink or show the source of a Reference (among other uses).
21	Uses Rating Table Object	This parameter has rating-dependent defaults and limits that can be obtained from the Rating Table Object. The Offline Read Full will include the default value for the smallest rating and limits that will accommodate the full range of values allowed in the family of devices using this particular combination of Family Code and Config Code. The Online Read Full will include the rating-dependent default and limit values for this particular combination of Family Code, Config Code, and Rating Code.
22	Writable Referenced Parameter	This bit must be zero unless the parameter is an Extended Parameter Reference. If the parameter is an Extended Parameter Reference, then: 0 = The referenced parameter may be read-only or writable. 1 = The referenced parameter must always be writable (including while running).
23	Disallow Zero	This bit must be zero unless the parameter is an Indirect Selector or Extended Parameter Reference. If the parameter is an Indirect Selector or Extended Parameter Reference, then: 0 = Allow zero 1 = Disallow zero If this bit is cleared (indicating that a value of zero is allowed), the device must support the “Zero Text” parameter attribute so that a software tool or HIM can obtain text from the Zero Text parameter attribute. If this bit is set (indicating that a value of zero is disallowed), a software tool or HIM will not allow the user to enter a value of zero.
24	Datalink Out	This bit is used by offline tools and indicates that this is a Datalink Out parameter. Bit 20 must also be set.
25	Datalink In	This bit is used by offline tools and indicates that this is a Datalink In parameter. Bits 20 and 22 must also be set.
26	Not Writable While IO Active	This parameter cannot be written if the I/O data being exchanged between the Host and the peripheral is valid.

Bit	Name	Description
27	Command Parameter	This parameter commands the drive to take an action, such as “Reset Defaults” or “Autotune,” and then returns to a value of zero. Offline software tools will not allow setting this parameter to anything other than a value of zero. If an offline file contains a Command Parameter with a non-zero value, the offline software tool will change the value to zero. Note that command parameters cannot have values that do not return to zero.
28	Current Value Is Default	This bit identifies a parameter that will not change if a “Reset Defaults” is commanded. For example, if a drive contains a Language parameter that is set to German, setting defaults will leave the parameter set to German. Likewise, if the parameter is set to French, setting defaults will leave the parameter set to French.
29	Use Zero Text	If the “Disallow Zero” bit is set, this bit must be cleared. If the “Disallow Zero” bit is cleared, then: 0 = Use Disabled Text parameter class attribute. 1 = Use Zero Text parameter instance attribute.
30-31	Reserved	Reserved

Formulas for Converting

Display Value = ((Internal Value + Offset) x Multiplier x Base) / (Divisor x $10^{\text{Decimal Places}}$)

Internal Value = ((Display Value x Divisor x $10^{\text{Decimal Places}}$) / (Multiplier x Base)) - Offset

Common Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Object Specific Services

Service Code	Implemented for:		Service Name	Allocation Size (in bytes)	
	Class	Instance		Par. Number	Par. Value
0x4B	Yes	No	Get_Attributes_Scattered	2	2
0x4C	Yes	No	Set_Attributes_Scattered	2	2
0x4D [1]	Yes	No	Get_Attributes_Scattered	4	4
0x4E [1]	Yes	No	Set_Attributes_Scattered	4	4

[1] These services are supported only when the adapter is used with a PowerFlex 750-Series drive.

The table below lists the parameters for the Get_Attributes_Scattered and Set_Attributes_Scattered object-specific service:

Name	Data Type	Description
Parameter Number	DWORD	Parameter to read or write
Parameter Value	DWORD	Parameter value to read or write (zero when reading)

DPI Fault Object

Class Code

Hexadecimal	Decimal
0x97	151

Products such as PowerFlex drives use this object for faults. Adapters use this object for events.

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Instances

The number of instances depends on the maximum number of faults or events supported in the queue. The maximum number of faults/events can be read in Instance 0, Attribute 2.

Instances (Hex.)	(Dec.)	Device
0x0000…0x3FFF	0…16383	Host
0x4000…0x43FF	16384…17407	Adapter
0x4400…0x47FF	17408…18431	DPI Port 1
0x4800…0x4BFF	18432…19455	DPI Port 2
0x4C00…0x4FFF	19456…20479	DPI Port 3
0x5000…0x53FF	20480…21503	DPI Port 4
0x5400…0x57FF	21504…22527	DPI Port 5
0x5800…0x5BFF	22528…23551	DPI Port 6
0x5C00…0x5FFF [1]	23552…24575	DPI Port 7
0x6000…0x63FF [1]	24576…25599	DPI Port 8
0x6400…0x67FF [1]	25600…26623	DPI Port 9
0x6800…0x6BFF [1]	26624…27647	DPI Port 10
0x6C00…0x6FFF [1]	27648…28671	DPI Port 11
0x7000…0x73FF [1]	28672…29695	DPI Port 12
0x7400…0x77FF [1]	29696…30719	DPI Port 13
0x7800…0x7BFF [1]	30720…31743	DPI Port 14

[1] These instances are supported only when the adapter is used with a PowerFlex 750-Series drive.

Example	Description
0	Class Attributes (Drive)
1	Most Recent Drive Fault
2	Second Most Recent Drive Fault
⋮	⋮
16384	Class Attributes (Adapter)
16385	Most Recent Adapter Event
⋮	⋮

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Class Revision	WORD	Revision of object
2	Get	Number of Instances	WORD	Maximum number of faults/events that the device can record in its queue
3	Set	Fault Command Write	BYTE	0 = No Operation 1 = Clear Fault/Event 2 = Clear Fault/Event Queue 3 = Reset Device
4	Get	Fault Trip Instance Read	WORD	Fault that tripped the device. For adapters, this value is always 1 when faulted.

Attribute ID	Access Rule	Name	Data Type	Description
5	Get	Fault Data List	STRUCT of: BYTE BYTE WORD[n]	Number of faults/events in the queue. A “0” indicates the fault queue is empty.
6	Get	Number of Recorded Faults	WORD	Number of faults/events in the queue. A “0” indicates the fault queue is empty.
7	Get	Fault Parameter Reference	WORD	Number of faults/events in the queue. A “0” indicates the fault queue is empty.

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Get	Full/All Information	STRUCT of WORD STRUCT of: BYTE BYTE STRING[16] STRUCT of: LWORD BOOL[16] WORD CONTAINER[n]	Fault code Fault source DPI port DPI Device Object Fault text Fault time stamp Timer value (0 = timer not supported) BOOL[0]: (0 = invalid data, 1 = valid data) BOOL[1]: (0 = elapsed time, 1 = real time) BOOL[2…15]: Not used Reserved Reserved
1	Get	Basic Information	STRUCT of WORD STRUCT of: BYTE BYTE STRUCT of: LWORD BOOL[16]	Fault code Fault source DPI port DPI Device Object Fault time stamp Timer value (0 = timer not supported) BOOL[0]: (0 = invalid data, 1 = valid data) BOOL[1]: (0 = elapsed time, 1 = real time) BOOL[2…15]: Not used
2	Get	International Fault Text	STRINGN	Text describing the fault with support for Unicode.

DPI Alarm Object

Class Code

Hexadecimal	Decimal
0x98	152

Products such as PowerFlex drives use this object for alarms or warnings. Adapters do not support this object.

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Instances

The number of instances depends on the maximum number of alarms supported by the queue. The maximum number of alarms can be read in Instance 0, Attribute 2.

Instances (Hex.)	(Dec.)	Device
0x0000…0x3FFF	0…16383	Host

Only host devices can have alarms.

Example	Description
0	Class Attributes (Drive)
1	Most Recent Alarm
2	Second Most Recent Alarm
⋮	⋮

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Class Revision	WORD	Revision of object
2	Get	Number of Instances	WORD	Maximum number of alarms that the device can record in its queue
3	Set	Alarm Command Write	BYTE	0 = No Operation 1 = Clear Alarm 2 = Clear Alarm Queue 3 = Reset Device
4	Get	Fault Data List	STRUCT of: BYTE BYTE WORD[n]	Reserved
5	Get	Number of Recorded Alarms	WORD	Number of alarms in the queue. A "0" indicates the alarm queue is empty.

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Get	Full/All Information	STRUCT of WORD	Alarm code
			STRUCT of:	Alarm source
			BYTE	DPI port
			BYTE	DPI Device Object
			STRING[16]	Alarm text
			STRUCT of:	Alarm time stamp
			LWORD	Timer value (0 = timer not supported)
			BOOL[16]	BOOL[0]: (0 = invalid data, 1 = valid data)
				BOOL[1]: (0 = elapsed time, 1 = real time)
				BOOL[2…15] Reserved
			WORD	Reserved
			CONTAINER[n]	Reserved
1	Get	Basic Information	STRUCT of WORD	Alarm code
			STRUCT of:	Alarm source
			BYTE	DPI port
			BYTE	DPI Device Object
			STRUCT of:	Alarm time stamp
			LWORD	Timer value (0 = timer not supported)
			BOOL[16]	BOOL[0]: (0 = invalid data, 1 = valid data)
				BOOL[1]: (0 = elapsed time, 1 = real time)
				BOOL[2…15] Reserved
2	Get	International Alarm Text	STRINGN	Text describing the alarm with support for Unicode.

DPI Diagnostic Object

Class Code

Hexadecimal	Decimal
0x99	153

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Instances

The number of instances depends on the maximum number of diagnostic items in the device. The total number of diagnostic items can be read in Instance 0, Attribute 2.

Instances (Hex.)	(Dec.)	Device
0x0000…0x3FFF	0…16383	Host
0x4000…0x43FF	16384…17407	Adapter
0x4400…0x47FF	17408…18431	DPI Port 1
0x4800…0x4BFF	18432…19455	DPI Port 2
0x4C00…0x4FFF	19456…20479	DPI Port 3
0x5000…0x53FF	20480…21503	DPI Port 4
0x5400…0x57FF	21504…22527	DPI Port 5
0x5800…0x5BFF	22528…23551	DPI Port 6
0x5C00…0x5FFF [(1)]	23552…24575	DPI Port 7
0x6000…0x63FF [(1)]	24576…25599	DPI Port 8
0x6400…0x67FF [(1)]	25600…26623	DPI Port 9
0x6800…0x6BFF [(1)]	26624…27647	DPI Port 10
0x6C00…0x6FFF [(1)]	27648…28671	DPI Port 11
0x7000…0x73FF [(1)]	28672…29695	DPI Port 12
0x7400…0x77FF [(1)]	29696…30719	DPI Port 13
0x7800…0x7BFF [(1)]	30720…31743	DPI Port 14

Example	Description
0	Class Attributes (Drive)
1	Drive Diagnostic Item 1
2	Drive Diagnostic Item 2
⋮	⋮
16384	Class Attributes (Adapter)
16385	Adapter Diagnostic Item1
⋮	⋮

[(1)] These instances are supported only when the adapter is used with a PowerFlex 750-Series drive.

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Class Revision	WORD	1
2	Get	Number of Instances	WORD	Number of diagnostic items in the device.
3	Get	ENUM Offset	WORD	DPI ENUM object instance offset

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Get	Full/All Information	STRUCT of:	
			BOOL[32]	Descriptor (see page C-19)
			CONTAINER [1]	Value
			CONTAINER	Minimum value
			CONTAINER	Maximum value
			CONTAINER	Default value
			WORD	Pad Word
			WORD	Pad Word
			STRING[4]	Units (for example, Amp, Hz)
			WORD	Multiplier [2]
			WORD	Divisor [2]
			WORD	Base [2]
			INT	Offset [2]
			DWORD	Link (source of the value) (0 = no link)
			STRING[16]	Diagnostic name text
1	Get/Set	Value	Various	Diagnostic item value
2	Get	International Diagnostic Item Text	STRUCT of:	
			STRINGN	Diagnostic name text
			STRINGN	Diagnostic units text
3	Get	International Full Read All	STRUCT of:	
			BOOL[32]	Descriptor
			CONTAINER	Value
			CONTAINER	Minimum
			CONTAINER	Maximum
			CONTAINER	Default
			WORD	Pad Word
			WORD	Pad Word
			WORD	Multiplier
			WORD	Divisor
			WORD	Base
			INT	Offset
			DWORD	Pad
			BOOL[32]	Extended descriptor
			STRINGN	Diagnostic name text
			STRINGN	Diagnostic units text

[1] A CONTAINER is a 32-bit block of data that contains the data type used by a value. If signed, the value is sign extended. Padding is used in the CONTAINER to ensure that it is always 32-bits.

[2] This value is used in the formulas used to convert the value between display units and internal units. Refer to Formulas for Converting on page C-21.

DPI Time Object

Class Code

Hexadecimal	Decimal
0x9B	155

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Instances

The number of instances depends on the number of timers in the device. Instance 1 is always reserved for a real-time clock although a device may not support it. The total number of timers can be read in Instance 0, Attribute 2.

Instances (Hex.)	(Dec.)	Device
0x0000…0x3FFF	0…16383	Host
0x4000…0x43FF	16384…17407	Adapter
0x4400…0x47FF	17408…18431	DPI Port 1
0x4800…0x4BFF	18432…19455	DPI Port 2
0x4C00…0x4FFF	19456…20479	DPI Port 3
0x5000…0x53FF	20480…21503	DPI Port 4
0x5400…0x57FF	21504…22527	DPI Port 5
0x5800…0x5BFF	22528…23551	DPI Port 6
0x5C00…0x5FFF [(1)]	23552…24575	DPI Port 7
0x6000…0x63FF [(1)]	24576…25599	DPI Port 8
0x6400…0x67FF [(1)]	25600…26623	DPI Port 9
0x6800…0x6BFF [(1)]	26624…27647	DPI Port 10
0x6C00…0x6FFF [(1)]	27648…28671	DPI Port 11
0x7000…0x73FF [(1)]	28672…29695	DPI Port 12
0x7400…0x77FF [(1)]	29696…30719	DPI Port 13
0x7800…0x7BFF [(1)]	30720…31743	DPI Port 14

Example	Description
0	Class Attributes (Drive)
1	Real Time Clock (Predefined) (not always supported)
2	Timer 1
3	Timer 2
⋮	⋮

[(1)] These instances are supported only when the adapter is used with a PowerFlex 750-Series drive.

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Class Revision	WORD	Revision of object
2	Get	Number of Instances	WORD	Number of timers in the object, excluding the real-time clock that is predefined.
3	Get	First Device Specific Timer	WORD	Instance of the first timer that is not predefined.
4	Set	Time Command Write	BYTE	0 = No Operation 1 = Clear all timers (Does not clear the real-time clock or read only timers)
5	Get	Number of Supported Time Zones	WORD	Number of time zones described in the Time Zone List attribute.
6	Get	Time Zone List	STRUCT	Identifies a time zone.
7	Get/Set	Active Time Zone ID	WORD	The ID field of the Time Zone List structure for the desired time zone.

Attribute ID	Access Rule	Name	Data Type	Description
8	Get	Active Time Zone Data	STRUCT of:	
			INT	Standard bias
			BYTE	Standard month
			BYTE	Standard day of week
			BYTE	Standard week
			BYTE	Standard hour
			BYTE	Standard minute
			BYTE	Standard second
			INT	Daylight offset
			BYTE	Daylight month
			BYTE	Daylight day of week
			BYTE	Daylight week
			BYTE	Daylight hour
			BYTE	Daylight minute
			BYTE	Daylight second
9	Get/Set	Custom Time Zone Data	STRUCT of:	
			INT	Standard bias
			BYTE	Standard month
			BYTE	Standard day of week
			BYTE	Standard week
			BYTE	Standard hour
			BYTE	Standard minute
			BYTE	Standard second
			INT	Daylight offset
			BYTE	Daylight month
			BYTE	Daylight day of week
			BYTE	Daylight week
			BYTE	Daylight hour
			BYTE	Daylight minute
			BYTE	Daylight second

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Get	Read Full	STRUCT of:	
			STRING[16]	Name of the timer
			LWORD or STRUCT	Elapsed time in milliseconds unless timer is a real-time clock (see attribute 2)
			BOOL[16]	See attribute 3
1	Get	Timer Text	STRING[16]	Name of the timer
2	Get/Set	Timer Value	LWORD -or-	Elapsed time in milliseconds unless the timer is a real-time clock.
			STRUCT of:	Real-Time Clock Data:
			WORD	Milliseconds (0…999)
			BYTE	Seconds (0…59)
			BYTE	Minutes (0…59)
			BYTE	Hours (0…23)
			BYTE	Days (1…31)
			BYTE	Months (1 = January, 12 = December)
			BYTE	Years (since 1972)
3	Get	Timer Descriptor	BOOL[16]	BOOL[0]: (0 = invalid data, 1 = valid data)
				BOOL[1]: (0 = elapsed time, 1 = real time)
				BOOL[2…15]: Not used
4	Get	International Read Full	STRUCT of:	
			STRINGN	International timer text
			STRUCT	Timer value
			BOOL[16]	Timer descriptor
5	Get	International Timer Text	STRINGN	Name of this timer
6	Get	Clock Status	BOOL[32]	Identifies clock status
8	Get/Set	Number of Leap Seconds	INT	Identifies the current number of Leap Seconds.
9	Get	Clock Options	BOOL[32]	Identifies the optional functionality available in the device's System Clock.
10	Get/Set	Clock Options Enable	BOOL[32]	Identifies which of the clock's options are enabled.

Host DPI Parameter Object

Class Code

Hexadecimal	Decimal
0x9F	159

To access Device parameters, use the DPI Parameter Object (Class Code 0x93).

Important: The Host DPI Parameter Object is supported only when the adapter is used with a PowerFlex 750-Series drive.

Instances

The number of instances depends on the number of parameters in the device. The total number of parameters can be read in Instance 0, Attribute 0.

Instances (Hex.)	(Dec.)	Device
0x0000…0x3FFF	0…16383	Reserved
0x4000…0x43FF	16384…17407	Adapter
0x4400…0x47FF	17408…18431	DPI Port 1
0x4800…0x4BFF	18432…19455	DPI Port 2
0x4C00…0x4FFF	19456…20479	DPI Port 3
0x5000…0x53FF	20480…21503	DPI Port 4
0x5400…0x57FF	21504…22527	DPI Port 5
0x5800…0x5BFF	22528…23551	DPI Port 6
0x5C00…0x5FFF	23552…24575	DPI Port 7
0x6000…0x63FF	24576…25599	DPI Port 8
0x6400…0x67FF	25600…26623	DPI Port 9
0x6800…0x6BFF	26624…27647	DPI Port 10
0x6C00…0x6FFF	27648…28671	DPI Port 11
0x7000…0x73FF	28672…29695	DPI Port 12
0x7400…0x77FF	29696…30719	DPI Port 13
0x7800…0x7BFF	30720…31743	DPI Port 14

Example	Description
16384	Class Attributes (Adapter)
16385	Adapter Parameter 1 Attributes
16386	Adapter Parameter 2 Attributes
⋮	⋮
17408	Class Attributes (HIM)
17409	HIM Parameter 1 Attributes
17410	HIM Parameter 2 Attributes
⋮	⋮

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Get	Number of Instances	WORD	Number of parameters in the device
1	Set	Write Protect Password	WORD	0 = Password disabled n = Password
2	Set	NVS Command Write	BYTE	0 = No Operation 1 = Store values in active memory to NVS 2 = Load values in NVS to active memory 3 = Load default values to active memory
3	Get	NVS Parameter Value Checksum	WORD	Checksum of all parameter values in a user set in NVS
4	Get	NVS Link Value Checksum	WORD	Checksum of parameter links in a user set in NVS
5	Get	First Accessible Parameter	WORD	First parameter available if parameters are protected by passwords. A “0” indicates all parameters are protected.
7	Get	Class Revision	WORD	2 = DPI
8	Get	First Parameter Processing Error	WORD	The first parameter that has been written with a value outside of its range. A “0” indicates no errors.
9	Set	Link Command	BYTE	0 = No Operation 1 = Clear All Parameter Links (This does not clear links to function blocks.)

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
6	Get	DPI Offline Read Full	STRUCT of:	
			BOOL[32]	Descriptor
			CONTAINER	Offline Minimum value
			CONTAINER	Offline Maximum value
			CONTAINER	Offline Default value
			STRING[16]	Parameter name
			STRING[4]	Offline parameter units
			WORD	Online minimum parameter instance
			WORD	Online maximum parameter instance
			WORD	Online default parameter instance
			WORD	Multiplier parameter instance
			WORD	Divisor parameter instance
			WORD	Base parameter instance
			WORD	Offset parameter instance
			BYTE	Formula number
			BYTE	Pad byte (always zero)
			WORD	Help instance
			WORD	Pad word (always a value of zero)
			CONTAINER	Parameter value
			WORD	Multiplier
			WORD	Divisor
			WORD	Base
			INT	Offset
7	Get	DPI Online Read Full	STRUCT of:	
			BOOL[32]	Descriptor (see page C-33)
			CONTAINER[1]	Parameter value
			CONTAINER	Minimum value
			CONTAINER	Maximum value
			CONTAINER	Default value
			WORD	Next parameter
			WORD	Previous parameter
			STRING[4]	Units (for example, Amps, Hz)
			WORD	Multiplier [2]
			WORD	Divisor [2]
			WORD	Base [2]
			INT	Offset [2]
			BYTE[3]	Link (source of the value) (0 = no link)
			BYTE	Always zero (0)
			STRING[16]	Parameter name
8	Get	DPI Descriptor	BOOL[32]	Descriptor (see page C-33)
9	Get/Set	DPI Parameter Value	Various	Parameter value in NVS. [3]
10	Get/Set	DPI RAM Parameter Value	Various	Parameter value in temporary memory. Valid only for DPI drives.
11	Get/Set	DPI Link	BYTE[3]	Link (parameter or function block that is the source of the value) (0 = no link)
12	Get	Help Object Instance	WORD	ID for help text for this parameter
13	Get	DPI Read Basic	STRUCT of:	
			BOOL[32]	Descriptor (see page C-33)
			CONTAINER	Parameter value
			CONTAINER	Minimum value
			CONTAINER	Maximum value
			CONTAINER	Default value
			STRING[16]	Parameter name
			STRING[4]	Units (for example, Amps, Hz)
14	Get	DPI Parameter Name	STRING[16]	Parameter name
15	Get	DPI Parameter Alias	STRING[16]	Customer supplied parameter name.
16	Get	Parameter Processing Error	BYTE	0 = No error 1 = Value is less than the minimum 2 = Value is greater than the maximum

Attribute ID	Access Rule	Name	Data Type	Description
18	Get	International DPI Offline Parameter Text	Struct of:	
			STRINGN	International parameter name
			STRINGN	International offline units
19	Get	International DPI Online Parameter Text	Struct of:	
			STRINGN	International parameter name
			STRINGN	International online units
20	Get	International DPI Online Read Full	Struct of:	
			BOOL[32]	Descriptor
			CONTAINER	Parameter value
			CONTAINER	Online minimum value
			CONTAINER	Online maximum value
			CONTAINER	Online default value
			WORD	Next
			WORD	Previous
			WORD	Multiplier
			WORD	Divisor
			WORD	Base
			INT	Offset
			BYTE[3]	Link
			BYTE	Pad word (always zero)
			BOOL[32]	Extended descriptor
			STRINGN	International parameter name
			STRINGN	International online parameter units
21	Get	DPI Extended Descriptor	UDINT	Extended Descriptor (see page C-34)
22	Get	International DPI Offline Read Full	Struct of:	
			BOOL	Descriptor
			CONTAINER	Offline minimum value
			CONTAINER	Offline maximum value
			CONTAINER	Offline default value
			WORD	Online minimum parameter instance
			WORD	Online maximum parameter instance
			WORD	Online default parameter instance
			WORD	Multiplier parameter instance
			WORD	Divisor parameter instance
			WORD	Base parameter instance
			WORD	Offset parameter instance
			BYTE	Formula number
			BYTE	Pad word (always zero)
			WORD	Help instance
			WORD	Pad word (always a value of zero)
			CONTAINER	Parameter value
			WORD	Multiplier
			WORD	Divisor
			WORD	Base
			INT	Offset
			BOOL[32]	Extended DPI descriptor
			STRINGN	International DPI parameter name
			STRINGN	International DPI offline parameter units

[1] A CONTAINER is a 32-bit block of data that contains the data type used by a parameter value. If signed, the value is sign extended. Padding is used in the CONTAINER to ensure that it is always 32-bits.

[2] This value is used in the formulas used to convert the parameter value between display units and internal units. Refer to Formulas for Converting on page C-35.

[3] Do NOT continually write parameter data to NVS. Refer to the attention on page 6-1.

Descriptor Attributes

Bit	Name	Description
0	Data Type (Bit 1)	Right bit is least significant bit (0). 000 = USINT used as an array of Boolean 001 = UINT used as an array of Boolean 010 = USINT (8-bit integer) 011 = UINT (16-bit integer) 100 = UDINT (32-bit integer) 101 = TCHAR ((8-bit (not Unicode) or 16-bits (Unicode)) 110 = REAL (32-bit floating point value) 111 = Use bits 16, 17, 18
1	Data Type (Bit 2)	
2	Data Type (Bit 3)	
3	Sign Type	0 = unsigned 1 = signed
4	Hidden	0 = visible 1 = hidden
5	Not a Link Sink	0 = May be the sink end of a link 1 = May not be the sink end of a link
6	Not Recallable	0 = Recallable from NVS 1 = Not Recallable from NVS
7	ENUM	0 = No ENUM text 1 = ENUM text
8	Writable	0 = Read only 1 = Read/write
9	Not Writable When Enabled	0 = Writable when enabled (e.g., drive running) 1 = Not writable when enabled
10	Instance	0 = Parameter value is not a Reference to another parameter 1 = Parameter value refers to another parameter
11	Uses Bit ENUM Mask	This parameter instance supports the Bit ENUM Mask attribute. For more information, see the definition of the attribute.
12	Decimal Place (Bit 0)	Number of digits to the right of the decimal point. 0000 = 0 1111 = 15
13	Decimal Place (Bit 1)	
14	Decimal Place (Bit 2)	
15	Decimal Place (Bit 3)	
16	Extended Data Type (Bit 4)	Bit 16 is the least significant bit. 000 = Reserved 001 = UDINT used as an array of Boolean 010 = Reserved 011 = Reserved 100 = Reserved 101 = Reserved 110 = Reserved 111 = Reserved
17	Extended Data Type (Bit 5)	
18	Extended Data Type (Bit 6)	
19	Parameter Exists	Used to mark parameters that are not available to network tools.
20	Not Used	Reserved
21	Formula Links	Indicates the Formula Data is derived from other parameters.
22	Access Level (Bit 1)	A 3-bit field used to control access to parameter data.
23	Access Level (Bit 2)	
24	Access Level (Bit 3)	
25	Writable ENUM	ENUM text: 0 = Read Only, 1 = Read/Write
26	Not a Link Source	0 = May be the source end of a link 1 = May not be the source end of a link
27	Enhanced Bit ENUM	Parameter supports enhanced bit ENUMs.
28	Enhanced ENUM	Parameter supports enhanced ENUMs.
29	Uses DPI Limits Object	Parameter uses the DPI Limits Object. Intelligent offline tools make use of the Limits Object to select limits and units.
30	Extended Descriptor	Parameter uses Extended Descriptor bits, which can be obtained by reading the DPI Extended Descriptor attribute for this parameter.
31	Always Upload/Download	Parameter shall always be included in uploads and downloads.

Extended Descriptor Attributes

Bit	Name	Description
0	Indirect Mode	0 = Analog (selects entire parameters) 1 = Digital (selects individual bits within parameters)
1	Indirect Type 0	Analog input list (Instance 0xFFFF)
2	Indirect Type 1	Digital input list (Instance 0xFFFE)
3	Indirect Type 2	Feedback list (Instance 0xFFFD)
4	Indirect Type 3	Analog output list (Instance 0xFFFC)
5	Indirect Type 4	Digital output list (Instance 0xFFFB)
6	Indirect Type 5	Undefined (Instance 0xFFFA)
7	Indirect Type 6	Undefined (Instance 0xFFF9)
8	Indirect Type 7	Undefined (Instance 0xFFF8)
9	Indirect Type 8	Undefined (Instance 0xFFF7)
10	Indirect Type 9	Undefined (Instance 0xFFF6)
11	Indirect Type 10	Undefined (Instance 0xFFF5)
12	Indirect Type 11	Undefined (Instance 0xFFF4)
13	Indirect Type 12	Undefined (Instance 0xFFF3)
14	Indirect Type 13	Undefined (Instance 0xFFF2)
15	Indirect Type 14	Parameter-specific list
16	FP Max Decimals Bit 0	These four bits are used on REAL parameters only. They indicate the maximum number of decimal places to be displayed for small values. A value of 0 indicates to not limit the number of decimal places used.
17	FP Max Decimals Bit 1	
18	FP Max Decimals Bit 2	
19	FP Max Decimals Bit 1	
20	Extended Parameter Reference	0 = Not an Extended Parameter Reference 1 = Extended Parameter Reference An Extended Parameter Reference contains a reference to another parameter. The value is formatted the same as an analog mode Indirect Selector parameter (SSpppp, where SS = slot number of device to which this Extended Parameter Reference is pointing, and pppp = number of the parameter or diagnostic item to which this Extended Parameter Reference is pointing). Note that an Extended Parameter Reference can only select parameters unlike an Indirect Selector. An Extended Parameter Reference could be used to configure a Datalink or show the source of a Reference (among other uses).
21	Uses Rating Table Object	This parameter has rating-dependent defaults and limits that can be obtained from the Rating Table Object. The Offline Read Full will include the default value for the smallest rating and limits that will accommodate the full range of values allowed in the family of devices using this particular combination of Family Code and Config Code. The Online Read Full will include the rating-dependent default and limit values for this particular combination of Family Code, Config Code, and Rating Code.
22	Writable Referenced Parameter	This bit must be zero unless the parameter is an Extended Parameter Reference. If the parameter is an Extended Parameter Reference, then: 0 = The referenced parameter may be read-only or writable. 1 = The referenced parameter must always be writable (including while running).
23	Disallow Zero	This bit must be zero unless the parameter is an Indirect Selector or Extended Parameter Reference. If the parameter is an Indirect Selector or Extended Parameter Reference, then: 0 = Allow zero 1 = Disallow zero If this bit is cleared (indicating that a value of zero is allowed), the device must support the “Zero Text” parameter attribute so that a software tool or HIM can obtain text from the Zero Text parameter attribute. If this bit is set (indicating that a value of zero is disallowed), a software tool or HIM will not allow the user to enter a value of zero.
24	Datalink Out	This bit is used by offline tools and indicates that this is a Datalink Out parameter. Bit 20 must also be set.
25	Datalink In	This bit is used by offline tools and indicates that this is a Datalink In parameter. Bits 20 and 22 must also be set.
26	Not Writable While IO Active	This parameter cannot be written if the I/O data being exchanged between the Host and the peripheral is valid.

Bit	Name	Description
27	Command Parameter	This parameter commands the drive to take an action, such as “Reset Defaults” or “Autotune,” and then returns to a value of zero. Offline software tools will not allow setting this parameter to anything other than a value of zero. If an offline file contains a Command Parameter with a non-zero value, the offline software tool will change the value to zero. Note that command parameters cannot have values that do not return to zero.
28	Current Value Is Default	This bit identifies a parameter that will not change if a “Reset Defaults” is commanded. For example, if a drive contains a Language parameter that is set to German, setting defaults will leave the parameter set to German. Likewise, if the parameter is set to French, setting defaults will leave the parameter set to French.
29	Use Zero Text	If the “Disallow Zero” bit is set, this bit must be cleared. If the “Disallow Zero” bit is cleared, then: 0 = Use Disabled Text parameter class attribute. 1 = Use Zero Text parameter instance attribute.
30-31	Reserved	Reserved

Formulas for Converting

Display Value = ((Internal Value + Offset) x Multiplier x Base) / (Divisor x $10^{\text{Decimal Places}}$))

Internal Value = ((Display Value x Divisor x $10^{\text{Decimal Places}}$) / (Multiplier x Base)) - Offset

Common Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Object Specific Services

Service Code	Implemented for:		Service Name	Allocation Size (in bytes)	
	Class	Instance		Par. Number	Par. Value
0x4D	Yes	No	Get_Attributes_Scattered	4	4
0x4E	Yes	No	Set_Attributes_Scattered	4	4

The table below lists the parameters for the Get_Attributes_Scattered and Set_Attributes_Scattered object-specific service:

Name	Data Type	Description
Parameter Number	UDINT	Parameter to read or write
Parameter Value	UDINT	Parameter value to read or write (zero when reading)

TCP/IP Interface Object

Class Code

Hexadecimal	Decimal
0xF5	245

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single

Instances

The adapter supports one instance of the TCP/IP Interface object.

Number	Description
0	Class Attributes
1	Object Attributes

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Revision	WORD	The revision of this object

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Status of TCP/IP Network Interface	DWORD	0 = Not configured 1 = Valid configuration 2…15 = Reserved
2	Get	Configuration Capability	DWORD	Bit I Value (0 = False, 1 = True) 0 = Supports BOOTP 1 = DNS Client (able to resolve host names by query to DNS server) 2 = DHCP Client (able to obtain network configuration through DHCP) 3 = DHCP-DNS Update (able to send its host name in the DHCP request) 4 = Configuration Settable (able to set the network configuration via TCP/IP) 5…31 = Reserved
3	Set	Configuration Control	DWORD	Bit I Value 1…3 = Startup configuration 0 = Use configuration saved in NVS 1 = Obtain configuration via BOOTP 2 = Obtain configuration via DHCP 3…15 = Reserved 4 = DNS Enabled (resolves host names by query to DNS server) 5…31 = Reserved
4	Get	Physical Link Object	STRUCT of: WORD Padded EPATH	 Path size Path
5	Get	Interface Configuration	STRUCT of: DWORD DWORD DWORD DWORD DWORD STRING	 Adapter's IP address Adapter's subnet mask Adapter's gateway address Primary name server Secondary name server Default domain name
6	Get	Host Name	STRING	Host name when using DHCP

Ethernet Link Object

Class Code

Hexadecimal	Decimal
0xF6	246

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x4C	No	Yes	Get_and_Clear

Instances

The adapter supports one instance of the TCP/IP Interface object.

Number	Description
0	Class Attributes
1	Object Attributes

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Revision	WORD	The revision of this object

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Interface Speed	DWORD	Speed in megabits per second (Mbs)
2	Get	Interface Flags	DWORD	Bit I Value 0 = Link status (0 = inactive, 1 = active) 1 = Duplex (0 = half duplex, 1 = full duplex) 2…31 = Reserved
3	Get	Physical Address	BYTE[6]	MAC address (XX-XX-XX-XX-XX-XX) The first octet (USINT[0]) is on the left.
4	Get	Interface Counters	STRUCT of: DWORD DWORD DWORD DWORD DWORD DWORD DWORD DWORD DWORD DWORD DWORD	 Octets received Unicast packets received Non-unicast packets received Inbound packets received but discarded Inbound packets with errors (not discarded) Inbound packets with unknown protocol Octets sent Unicast packets sent Non-unicast packets sent Outbound packets discarded Outbound packets with errors

Attribute ID	Access Rule	Name	Data Type	Description
5	Get	Media Counters	STRUCT of:	RX = Received, TX = Transmitted
			DWORD	RX frames not having integral number of octets long
			DWORD	RX frames not passing FCS check
			DWORD	TX frames having one collision
			DWORD	TX frames having multiple collisions
			DWORD	Number of times of SQE test error message
			DWORD	TX Frames delayed first attempt by busy medium
			DWORD	Collisions detected later than 512 bit-times in trans.
			DWORD	TX frames failing due to excessive collisions
			DWORD	TX frames failing due to intern MAC sublayer TX error
			DWORD	Times of carrier sense condition loss during trans.
			DWORD	RX frames exceeding the maximum frame size
			DWORD	RX frames failing due to intern MAC sublayer RX error